MOBILE UM MAP
(CHAT ROOM MODULE)

CHEW SIONG KHING
WEK 030034
SUPERVISOR: PN. NAZEAN JOMHARI
MODERATOR: PN. AZWINA

FAKULTI SAINS KOMPUTER DAN
TEKNOLOGI MAKLUMAT
UNIVERSITI MALAYA
SESI 2005 / 2006

ABSTRACT

In this project, the system that being developed is called Mobile UM Map. There are two main parts of the system, which are map system and chat room system. The map system will allow the users to browse the University of Malaya (UM) map and search the path from one location to another location within the campus. While the chat room system allow the multi-users to carry out the discussion on the current issues that occurred within the campus. The whole application will be running in the mobile phone which fulfills the minimum requirements which is Nokia Series 60.

In the whole report, all the explanations and descriptions will be emphasizes on the chat room module. The objective of the chat room module in the Mobile UM Map is to allow the user to familiarize the UM environments. By using the chat room system in the application, the UM students and staffs will be able to keep alert on the latest news or activities that carried out within the campus. They will be able to feel that they are part of the family in the UM campus.

The Mobile UM Map is the mobile application which will consists of mobile client, server and database. In developing the Mobile UM Map, the programming language that used to develop the mobile client is J2ME. For development of the server part is J2EE Servlet and the database that being chosen is MySQL database. The tool that used to develop the whole application is Eclipse.

ACKNOWLEDGEMENT

In developing the chat room module, I had been faced some difficulties and problems. However, all of this problems was not a barriers for me to complete the application because there are a few of people willing to help me when I face the problems. I would like to take the opportunity to express my gratitude to them.

The first person that I would like to thanks is Pn. Nazean who is my project supervisor. She did give a lot of guidance and advices for me and give the opportunity for me to take part in the exhibition in the Malaysia Expo Technology (MTE).

The second person that I like to thanks is my first panel in WXES 3181, Pn. Azlina. During the presentation of the WXES3181, she had given some comments to me on the problems that should be considered when developed the application.

I would also like to express my gratitude to my main panel, Pn. Azwina and my second panel Ms. Su Moon Ting. They also had given their valuable comments that the chat room module should include. Their advices would be my improvements in the future.

Apart from that, I also feel grateful with my group leader, Lee Kok Siong who had been guide me and share a lot of his knowledge and experiences with me during developing the Mobile UM Map.

5.0	System Analysis	45
	5.1 Functional Requirements	45
	5.2 Non-functional Requirements	51
	5.3 Hardware and Software Requirements	52
6.0	System Design	55
	6.1 System Architecture	55
	6.2 Interface Layer Design	56
	6.3 Data Processing Layer Design	57
	6.4 Data Collection Design	58
	6.5 Procedural Design	59
	6.6 Data Design	65
7.0	System Development	66
	7.1 Database Development	66
	7.2 Mobile Client Development	68
	7.3 Server Development	71
	7.4 Code Example	72
8.0	System Testing	88
	8.1 Component Testing	88
	8.2 System Testing	89

9.0	Discussion	90
	9.1 Problems and Solutions	90
	9.2 User Evaluation	90
	9.3 Positive and Negative Features	90
	9.4 Future Enhancements	91
10.0	Conclusion	92
11.0	References	93

LIST OF TABLES

TABLE	TITLE	PAGE NUMBER
2.1	Comparison of Mobile Chat room and Computer Chat room	31
3.1	Table show the instruction of services	34
5.1	Register Use Case Description	46
5.2	Login Use Case Description	46
5.3	View room list Use Case Description	47
5.4	View member list Use Case Description	48
5.5	Send message Use Case Description	49
5.6	Receive message Use Case Description	49
5.7	Logout Use Case Description	50

6.1	User Table Design	65
6.2	Room Table Design	65
8.1	Login module test data	88
8.2	Test data for Registration module	88
8.3	Test data for chat room module	89

LIST OF FIGURES

FIGURE	TITLE	PAGE NUMBER
1.1	UM Map	1
1.2	Spiral Model of software process	7
1.3	Gantt Chart	9
2.1	Client and Server components	17
2.2	Secure client/server interactions	19
2.3	The Threaded Chat user interface	24
3.1	Architecture of Viztel's BuddyTalk	32
3.2	The web that show the status of chat users	32
3.3	The process of inviting chat friends	33
3.4	Screenshot of Mobile Chat Room	34

3.5	MoCha Screenshot	41
5.1	Use Case of Chat room module	45
5.2	Layout of Eclipse	52
5.3	Layout of MySQL Query Browser	54
6.1	System architecture of Mobile UM Map	55
6.2	Interface layer design	56
6.3	Data Processing Layer Design	57
6.4	Data Collection Layer Design	58
6.5	Login Sequence Diagram	59
	Register Sequence Diagram	60
6.6	View Room List Sequence Diagram (From Login Form)	61

1.0 INTRODUCTION

1.1 PROJECT OVERVIEW

University of Malaya has 16 faculties, 12 College within campus and a few others of buildings such as library, DTC, and Gymnasium. The details of the location of each building are shown in the UM map as Figure 1.1. Thus, it is necessary to provide UM students and UM staffs with a UM map especially for those first years students, new staffs and visitors.

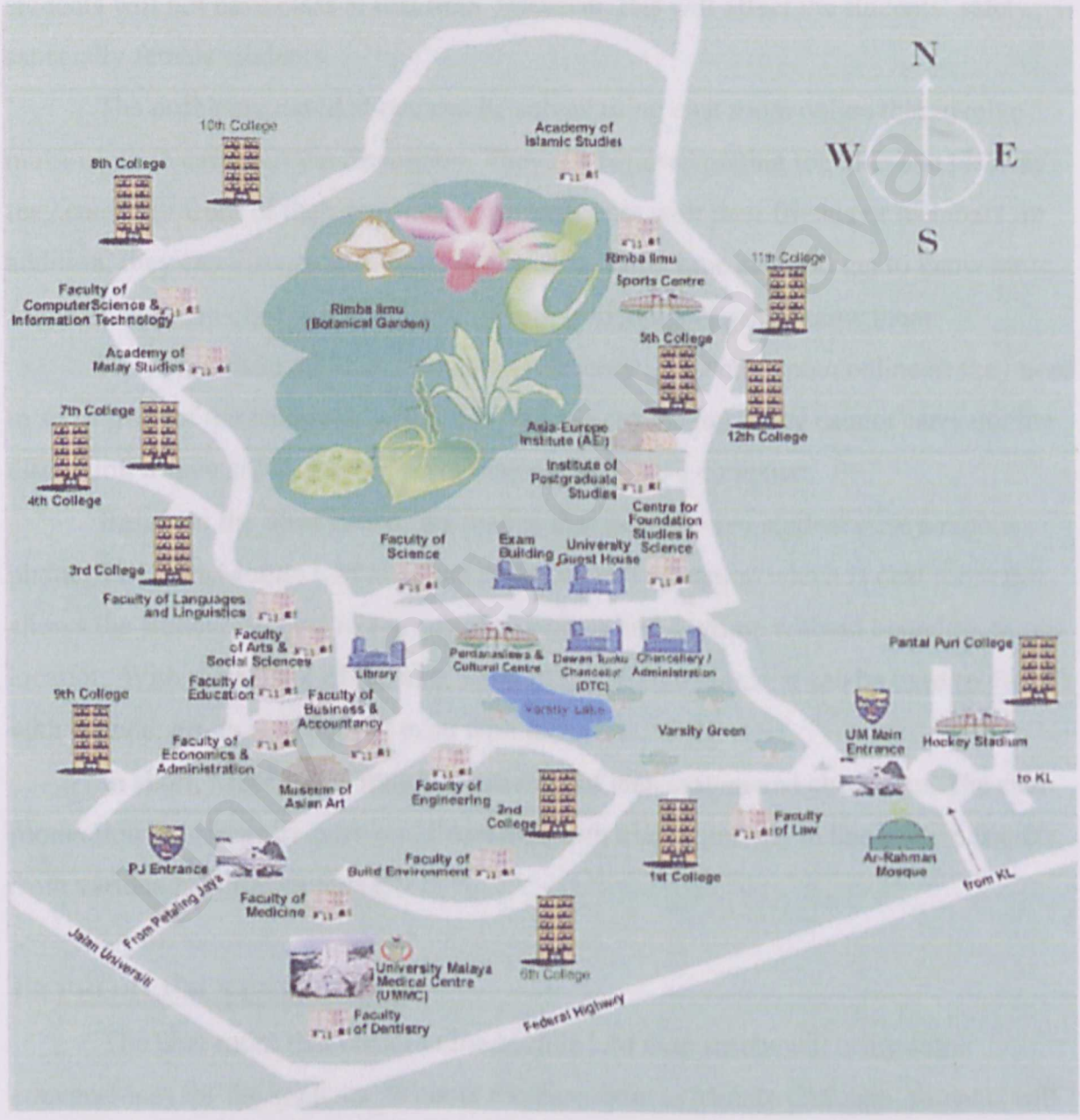


Figure 1.1: UM map (Source: http://um.edu.my, obtained September 20, 2005)

In the university lifestyle, it is also very important for a student to be social able because there will be a lot of task that required the group discussion to be carried out. Normally, they will meet at a place together to conduct the discussion. This will raise a few problems such as time consuming because need to wait until every members to reach the place before the discussion can be carried out. Apart from that, the time to carry out the discussion is limited because each of them will have class or other activities. Sometime, they will carry out the discussion at night because most of the students will not have class at that time. However, this will affect the students' safety especially female students.

The problems stated above can be solved using chat room online that involve multi-users to carry out the discussion. They not required getting together at a place as they can sit in front of their own computer to discuss with their friends or members. In addition, they can discuss with other group at the same time and also get to know more friend as they can choose any of the chat rooms to join and get to know them.

However, there still has some inconveniency using chat room online as they need to sit in front of the computer which has Internet connection. They cannot carry out the discussion if they are at another place where didn't has a computer.

Based on the observation, we realize that almost every student pose a mobile phone. Thus, the Mobile UM map had added an extra function which is chat room that allows the students to chat in a group at anywhere and anytime without boundary to one location. With the advent of the chat room in the Mobile phone, it can be used to discuss with friends, group members or even with lecturers.

In short, Mobile UM map is consisting of map system and chat room. The chat room allows students to carry out discussion with friends and get to know more friends from various faculties or college.

1.2 PROBLEM STATEMENT

The chat room that install in the Mobile UM map surely will bring some conveniences for the students. Without the chat room in Mobile UM map, students will face the following problems or inconveniency:

- Students need to meet face to face to conduct discussion.

- The discussion that carried out might not comfortable for students if the place is hot.
- Just can conduct the discussion at certain times only as everyone will have others activities.
- Cannot conduct discussion with different group at the same times.
- The noise made during face to face discussion will affect the listening of peoples' ideas.

Mobile UM map will solve the problems above by providing the chat room for them to carry out the discussion at mobility way as the mobile phone is a ubiquity device.

In developing the Mobile Map, there are a few issue need to be considered. One of the issues is the security of the users. Since the main concern in the chat room system is to identify who are the people in the room. That's why it is important to authenticate the users who login to the chat room in Mobile UM map.

The second issue in the chat room system is the problems raised when using the system such as difficulties to differentiate users easily who are posting the message. However, this can be solved by using different color in text to represent the different user.

Since the time is very important to the students, it is ideal that the Mobile UM map allows students to carry out discussion without required spending times to walk to a place to gather together with friends.

1.3 PROJECT OBJECTIVES

1.3.1 General Objectives

1) Allow the students to familiarize the environments in UM including the buildings, people, issues, activities or any incidents occurring within the campus.
2) Allow students to carry out discussion with friends at anytime and anywhere without boundary to one location.
3) Utilize the mobile phone as a mobility device to carry out the discussion.
4) Allow students to carry out the discussion with different group at the same times.

1.3.2 Specific Objectives

1) Provide the opportunities for students to get to know more friends.
2) Allow students to communicate with other peoples during leisure times.
3) Strengthens the relationships among students.
4) Ensure the safety of the students if the discussion carried out at night.
5) Save times of students from requiring reaching at a place together.
6) Allow students to discuss with friends even during holidays when most of them are at their home town respectively.
7) Carry out the discussion at the comfortable way since they no need to gather together at the noisy place.
8) Ensure that the students will produce the fruitful idea under comfortable conditions.
9) Practice students to be well organized in the discussion when they carry out discussion with different group simultaneously.
10) Can take note of the discussion easily by looking at the message on the phone screen.
11) Act as the mobile counseling because the students can share their problems with the other students using the chat room and seek for the advices from other people.
12) Encourage students to share their knowledge and experience with other students.

1.4 PROJECT SCOPE

The project is to develop the Mobile UM Map that can substitute the traditional paper map. The Mobile UM map consists of two main functions which is UM map displayed, and also the chat room.

One of the module of system which is chat room is required the users to enter the text message. The chat room will allow multi-user to be joined in order to carry out the discussion among students.

The Mobile UM map will be implemented using J2ME to develop the Mobile client and J2EE to develop the server program. The HTTP connection will be adopted to allow the communication between the client and server. While for the database will be

developed using MySQL. Then the database will be connected to the J2EE Servlet in the Tomcat server.

The constraint of the Mobile UM map is the limited of the memory that the J2ME phone can support. That will raise some problem such as low speed and limited function to be implemented. However, the advent of the J2EE will reduce the burden of the J2ME device.

The target users of the Mobile UM is the UM students, UM staffs, and even visitors. The students and staffs especially first year student and junior staffs will use the Mobile UM map to check the location of particular destination in UM. They can also use the system to carry out the discussion with multi users.

1.5 DESCRIPTION OF METHODOLOGY

In order to develop the Mobile UM map, it is necessary to choose the suitable methodology to provide a more structure process in completing the mobile system. Since the development of Mobile UM map is responsible by two people, thus it is suitable to adopt the eXtreme programming. With the complement of spiral model, it will help to produce a more quality program.

Let delve deeper the characteristics of eXtreme programming:

- Code reviewed constantly
- Always carry out the test on the system
- The integration test will be carried out when the system being added new functions or modified
- Keep the code as simple as possible
- Shorter iterations

The eXtreme programming brings some beneficial such as:

- *Economics.* Event though the cost will increase by adding the second person during the development, but it will reduce the defect. The cost of defect removal will surely been saved a lot compare spending the extra expenses in placing the pair programmer in the code program.
- *Satisfaction.* It is more enjoyable to work in pair compare to working alone.

- *Design quality.* Working in pair can produce a better quality of codes or design because they will have different viewpoints and arguing the better results.
- *Continuous Reviews.* The pair programmer will hand in hand to review the code continuously to reduce the defect.
- *Problem Solving.* Pair programmers intend to solve the problems faster and effectively.
- *Learning.* Pair programmers will share the knowledge with each other from various types of skills and design technique.
- *Team Building and Communication.* Working in pair will improve the communication and team work effectively.
- *Staff and Project Management.* Over the projects, the skills of the team member will increase. This will reduce the risk of losing key programmer since there are other programmer had the related experience and knowledge.

[Alistair Cockburn, and Laurie Williams, 2000]

After review some of the characteristics of eXtreme programming, let look at the characteristics of spiral model. According to Barry Boehm, and Wilfred J.Hansen. (2001:2) the definition of Spiral Development model is:

> "*The spiral development model is a risk-driven process model generator that is used to guide multi-stakeholder concurrent engineering of software-intensive system. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.*"

The Figure 1.2 shows the original model of Spiral Model. As shown in the diagram, the initial step for each cycle of spiral is to identify the objectives, alternatives, and constraints.

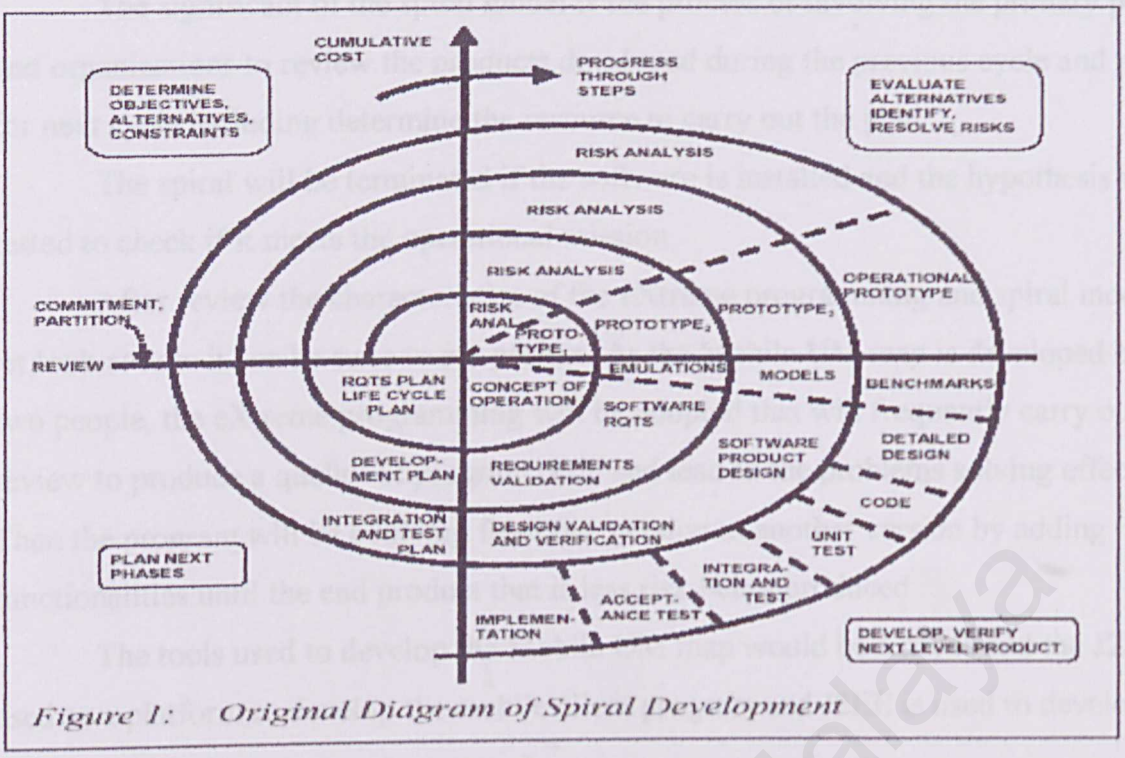


Figure 1: Original Diagram of Spiral Development

Figure 1.2: Spiral Model of software process

(Source: http://sunset.usc.edu/publications/TECHRPTS/2001/usccse2001-501/usccse2001-501.pdf, obtained September 9, 2005)

After the identification process being made, the next step would be evaluating the alternatives that relative to the objectives and constraints. In this process, it will normally lead to the identification of area uncertainty that is causing the project risks. Thus, some of the resolution techniques such as simulation, prototyping, and reference checking need to be chosen in order to resolve the source of project risks.

After that, determine the next step by the relative remaining risks. Let say if the user-interface risks will dominate the whole program development, the next step may can be the evolvement of development such as develop the more details prototype.

If the previous prototype had resolve the project risk, then the next step would be following the basic waterfall approach (concept of operation, software requirements, preliminary design, etc.) that been appropriate modified to incorporate the incremental development.

The significant of the spiral model is the process of involving the primary people and organizations to review the products developed during the previous cycle and plan for next cycle including determine the resource to carry out the plan.

The spiral will be terminated if the software is installed and the hypothesis was tested to check if it meets the operational mission.

After review the characteristics of the eXtreme programming and spiral model, let look at how it can be suite to my project. As the Mobile UM map is developed by two people, the eXtreme programming will be adopted that will frequently carry out the review to produce a quality of program code and lead to the problems solving effectively. Then the program will be evolving from one version to another version by adding the functionalities until the end product that is less risk being produced.

The tools used to develop the Mobile UM map would be Eclipse and the J2ME is used as a platform to develop the mobile client program and J2EE is used to develop the server program.

(Barry W. Boehm, 1988:64-65)

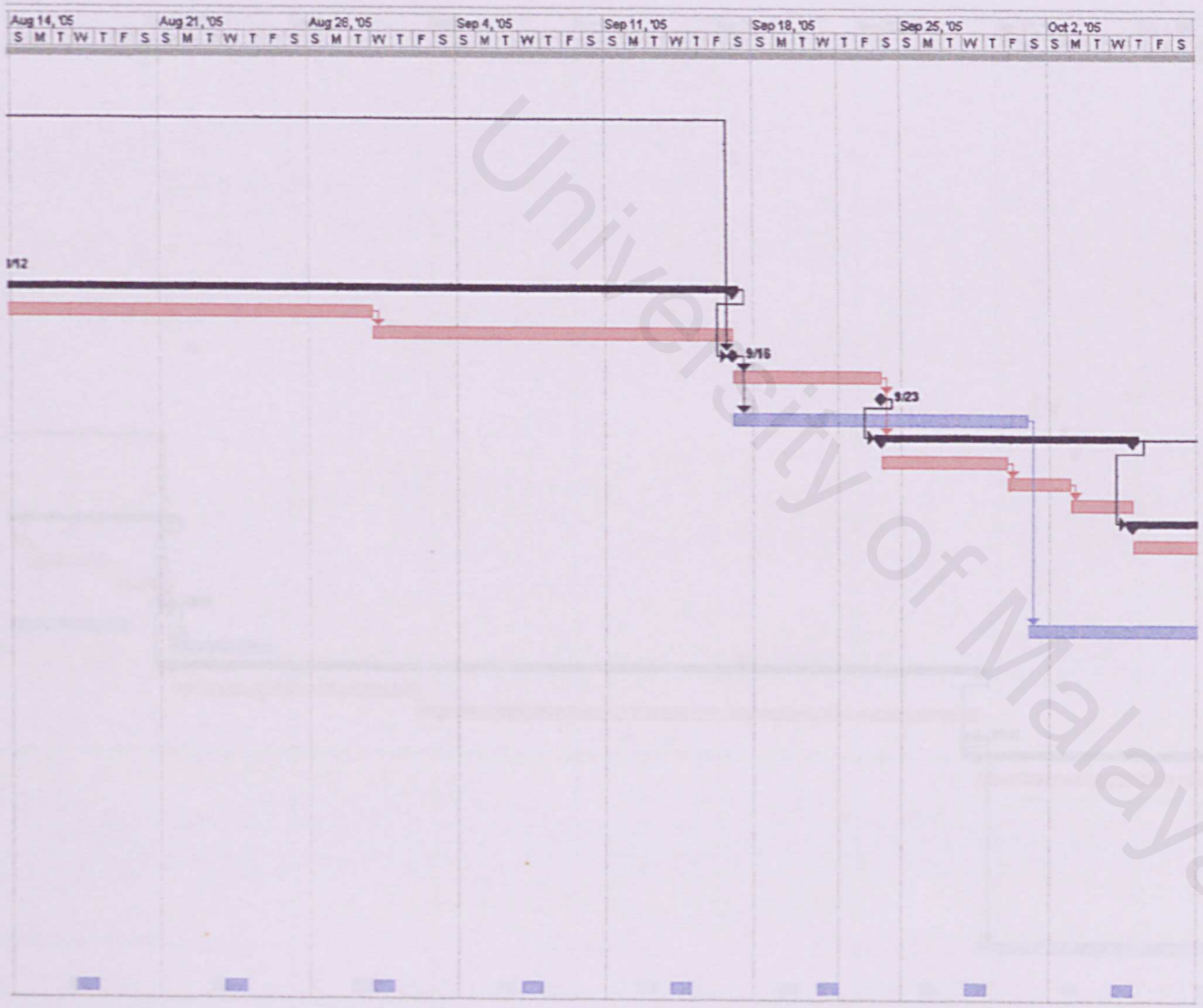
1.6 EXPECTED OUTCOME

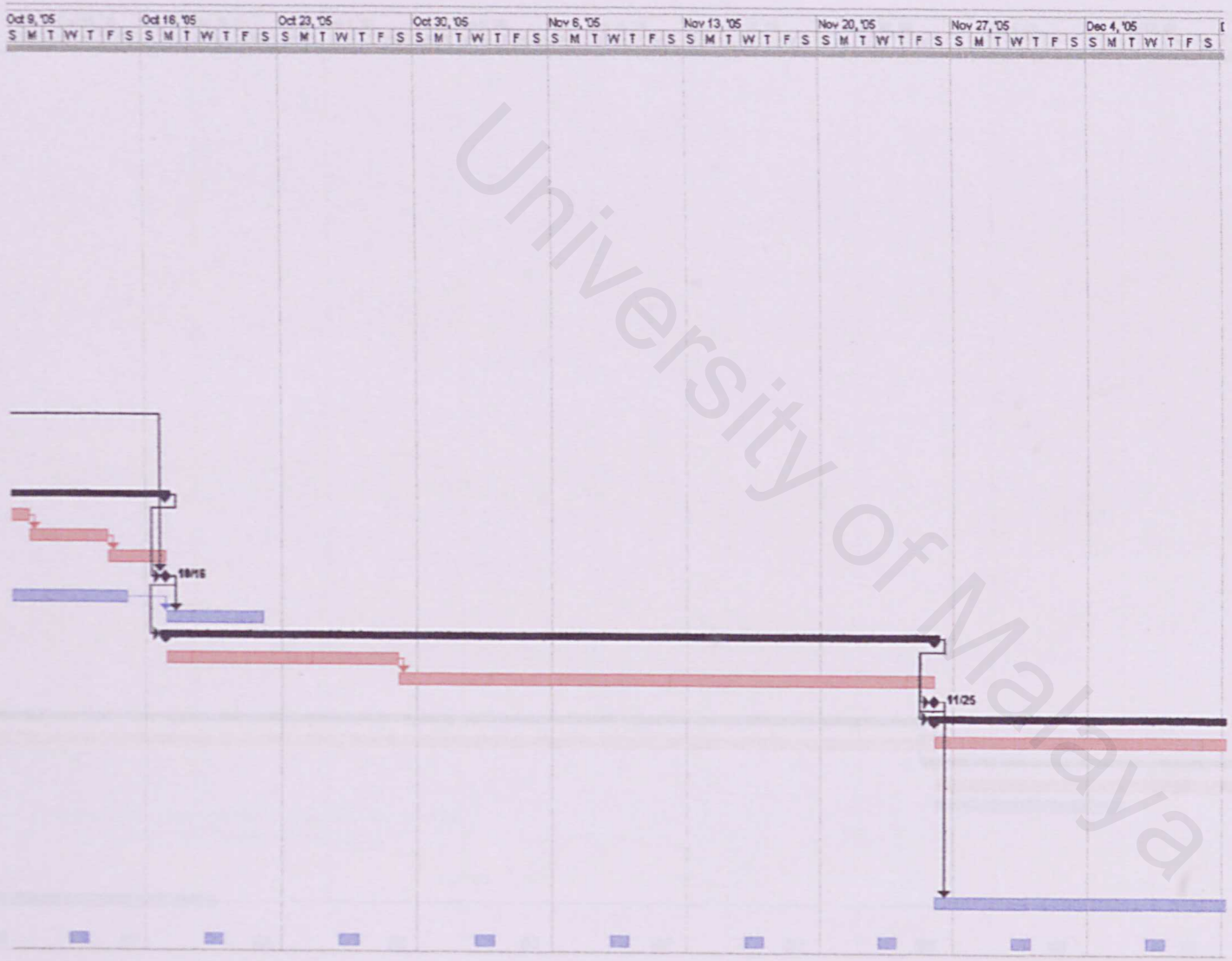
The project intend to create the Mobile UM map that install in the mobile phone that can allow users to search location using mobile UM map and mobile chat room.

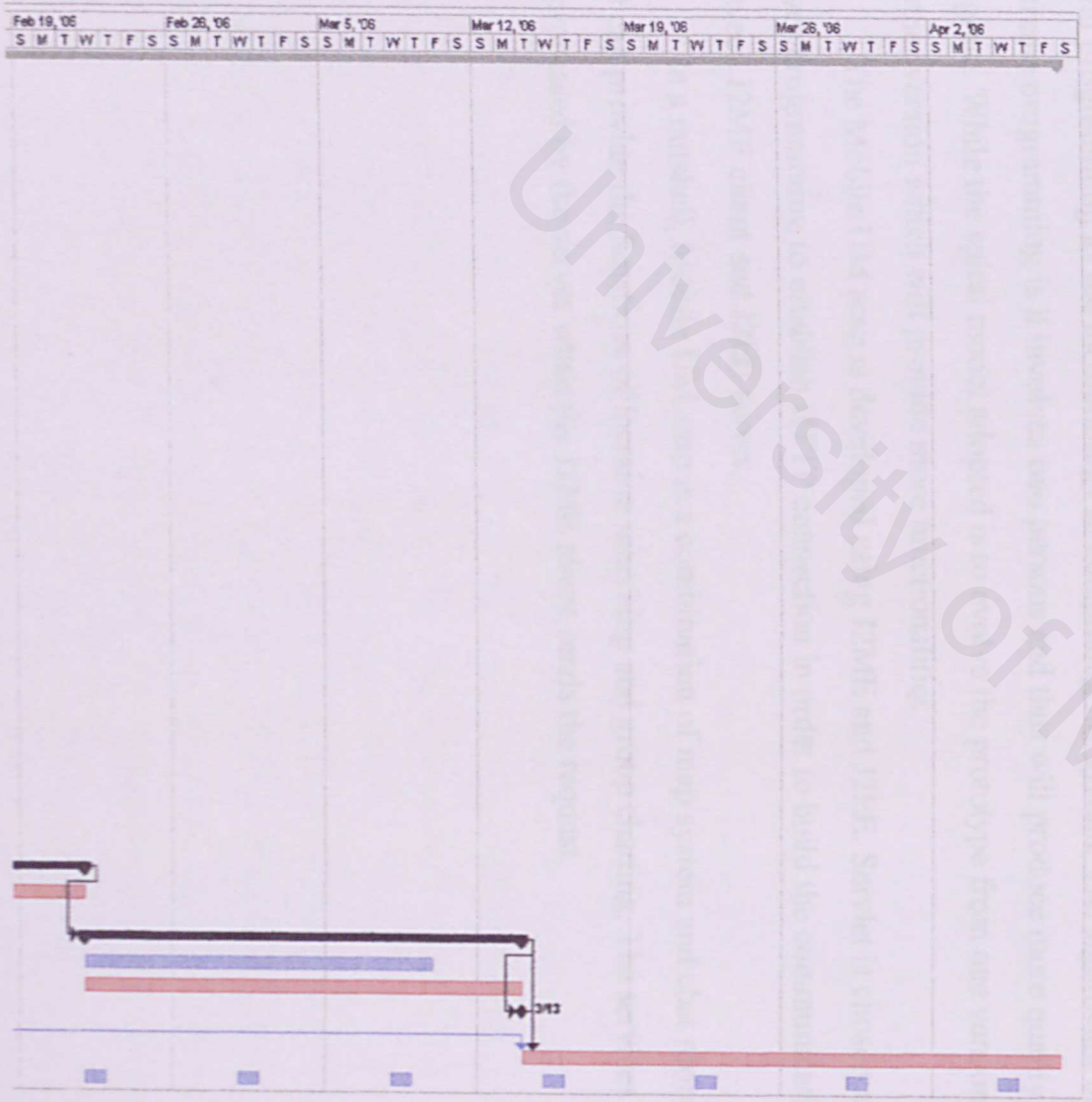
The project will produce a set of documentation including literature reviews, proposed solutions and requirements specifications, and design specification. The prototypes that being developed will be evolve from one version to another version by adding the extra functions.

1.7 PROJECT TIMELINE

ID	Task Name	Duration	Start	Finish
0	**Mobile UM Map**	########	##########	**Fri 4/7/06**
1	Choose Topic	5 days	Mon 7/11/05	Fri 7/15/05
2	Find Articles	21 days	Sat 7/16/05	Fri 8/5/05
3	**Chapter 1-Introductior**	**15 days**	**Sat 7/16/05**	**Sat 7/30/05**
4	Project Scope	15 days	Sat 7/16/05	Sat 7/30/05
5	Problem Statement	15 days	Sat 7/16/05	Sat 7/30/05
6	Project Objectivies	15 days	Sat 7/16/05	Sat 7/30/05
7	Project Realization	15 days	Sat 7/16/05	Sat 7/30/05
8	Submit Articles	0 days	Fri 8/5/05	Fri 8/5/05
9	Summary of Articles	7 days	Sat 8/6/05	Fri 8/12/05
10	Submit Summary of Articl	0 days	Fri 8/12/05	Fri 8/12/05
11	**Chapter 2-Literature R**	**35 days**	**Sat 8/13/05**	**Fri 9/16/05**
12	Background Study	18 days	Sat 8/13/05	Tue 8/30/05
13	Existing System	17 days	Wed 8/31/05	Fri 9/16/05
14	Submit Proposal	0 days	Fri 9/16/05	Fri 9/16/05
15	Tools Justification	7 days	Sat 9/17/05	Fri 9/23/05
16	Submit Tools Justification	0 days	Fri 9/23/05	Fri 9/23/05
17	Mastery of the Tools	14 days	Sat 9/17/05	Fri 9/30/05
18	**Chapter 3-Methodolog**	**12 days**	**Sat 9/24/05**	**Wed 10/5/05**
19	Research Methodolc	6 days	Sat 9/24/05	Thu 9/29/05
20	Algorithm	3 days	Fri 9/30/05	Sun 10/2/05
21	Methods	3 days	Mon 10/3/05	Wed 10/5/05
22	**Chapter 4-System Ana**	**11 days**	**Thu 10/6/05**	**Sun 10/16/05**
23	Functional Requirem	4 days	Thu 10/6/05	Sun 10/9/05
24	Non-Functional Req	4 days	Mon 10/10/05	Thu 10/13/05
25	Hardware & Softwa	3 days	Fri 10/14/05	Sun 10/16/05
26	Submit Requirements Spe	0 days	Sun 10/16/05	Sun 10/16/05
27	Prototyping	14 days	Sat 10/1/05	Fri 10/14/05
28	Viva	1 wk	Mon 10/17/05	Fri 10/21/05
29	**Chapter 5-System Des**	**40 days**	**Mon 10/17/05**	**Fri 11/25/05**
30	Modules	12 days	Mon 10/17/05	Fri 10/28/05
31	User Interface	29 days	Sat 10/29/05	Fri 11/25/05
32	Submit Design Specificati	0 days	Fri 11/25/05	Fri 11/25/05
33	**Chapter 6-System Imp**	**68 days**	**Sat 11/26/05**	**Wed 2/1/06**
34	Program Coding	68 days	Sat 11/26/05	Wed 2/1/06
35	**Chapter 7-System Tes**	**28 days**	**Thu 2/2/06**	**Tue 2/21/06**
36	Components	20 days	Thu 2/2/06	Tue 2/21/06
37	Whole System	10 days	Thu 2/2/06	Sat 2/11/06
38	**Chapter 8-Discussion**	**20 days**	**Wed 2/22/06**	**Mon 3/13/06**
39	Maintenance	18 days	Wed 2/22/06	Thu 3/9/06
40	Improvement	20 days	Wed 2/22/06	Mon 3/13/06
41	Submit Report	0 days	Mon 3/13/06	Mon 3/13/06
42	User Manual	30 days	Sat 11/26/05	Sun 12/25/05
43	Presentation	25 days	Tue 3/14/06	Fri 4/7/06
44	**Discussion With Super**	**250 days**	**Wed 7/26/06**	**Wed 4/5/06**







1.8 SUMMARY

The idea of developing the Mobile UM map is due to the reason of the need to ease the new UM students or staffs to get the particular locations in UM. The main purpose is to save the times and energies when need to reach a location where they never gone before. It also used to help students to carry out the discussion and tighten the relationships among the friends.

Actually, the Mobile UM map is implemented on the Mobile phone because it is a ubiquity device whereby almost everyone poses the mobile device. This will allow users to use the map system at anytime and anywhere without boundary to one location.

The significant of the Mobile UM map is that it provides the chat room to let multi-user to enter message in form of text in order to chat among a group of friends. It also allow users to play the UM map game to make them more familiar with the UM environments.

During the process of development, I choose the eXtreme programming and spiral programming as the methodology to completing my system. The significant of the eXtreme programming is it involves two persons and thus will produce more quality program. While the spiral model adopted is to evolve the prototype from one version to another version which will provide more functionalities.

The Mobile UM map is developed using J2ME and J2EE. Servlet is chosen as a server programming to establish HTTP connection in order to build the communication between J2ME client and J2EE server.

In a nutshell, Mobile UM map is a combination of map system and chat room that can provide the functions of location searching and group chatting. The services will be processed by the server when the J2ME client sends the request.

2.0 LITERACTURE REVIEW

2.1 J2ME END-TO-END SECURITY

2.1.1 Security Solutions for Different Layer

There are the securities solutions that being suggested, that are reside in the transport layer and application layer for end-to-end security. Each of the two layers will have different of security solutions.

2.1.1.1 Security Solutions in Transport Layer

The security protocol in transport layer can be further divided into two type of architecture. One of them is using the gateway to switch the non-standard Internet protocols in wireless network to standard Internet Protocol in wired network. The second architecture is without gateway and using the Internet Protocol in a way that can respect the resources limitation in the wireless device.

Security protocol with gateway

Wireless Application Protocol (WAP 1.x) is using the first architecture model which required the gateway to switch the protocol used in wireless network into protocol used in wired network. The security protocol adopted in WAP is Wireless Transport Layer Security Protocol (WTLS). The responsibilities of the WTLS is to secure the data transmit in wireless network between mobile phone and WAP gateway. While the Secure Socket Layer (SSL) protocol is used to secure the data transmit in wired network between the WAP gateway and web server. (Wassim Itani, &Ayman Kayssi, 2004:15)

The WAP 1.x is no end-to-end security provided because the data in the WAP gateway remains unencrypted during the switching protocol process. However, the WAP 2.0 had improved the security problems. WAP 2.0 does not required gateway because it adds support for the Internet protocol and this turn the WAP-enabled mobile phone into the Internet device. (Wassim Itani, &Ayman Kayssi, 2004:15)

The disadvantages of the upgraded WAP 2.0 is required the major changes for the mobile phone and wireless networks. Furthermore, WAP is a browser-based technology that not suitable to develop the application that required user fast interaction and complex client side logic. If compare to java-based application, the browser-based

application is larger in network traffic because the application is reside in the server and required the constant network availability. In addition, the use of XHTML instead of WML caused the network traffic further increased because of the richness tags and features enhanced that XHTML process. (Wassim Itani, &Ayman Kayssi, 2004:16)

Security protocol without gateway

The second architecture which uses the standard Internet protocol without gateway will adopt the famous security protocol in Internet that is SSL. SSL operate by generating the secure channel on top of TCP, and using the confidentiality, certificate, and message integrity to provide server authentication. However, due to the reason that the SSL is consuming a lot of memory, thus Sun Microsystems had develop KSSL which is light-weight SSL 3.0 implementation in the J2ME-device. (Wassim Itani, &Ayman Kayssi, 2004:16)

The advantages of the KSSL in J2ME device are allowing the direct and secure communication with a large number of web servers. Apart from that, the KSSL also increase the size of the J2ME virtual machine by roughly 25%. KSSL is recommended protocol that can support the HTTPs (HTTP over SSL) in the MIDP 2.0. (Wassim Itani, &Ayman Kayssi, 2004:17)

The disadvantages of the KSSL are the undesirable performance whereby the full handshake SSL operation need to be carried out every time communicates with new web server. Apart from that, the SSL will not be able to categorized the sensitivity of data during perform the encryption operation and thus using the same key-strength to encrypt all the data there is unnecessary for some wireless application. (Wassim Itani, &Ayman Kayssi, 2004:18)

2.1.1.2 Security Solution in Application Layer

In the application layer, the eXtensible Markup Language (XML) is used as a format for data communication between J2ME client and server. The XML file is secured when transfer from client to server by using XML security protocol. There are some XML security protocols such as XML encryption protocol which used to encrypt certain part all whole of the XML file using the symmetric ciphering techniques; XML digital signature protocol which specify the way to digitally sign the XML file; and

Security Assertion Markup Language (SAML) which used to embed the authentication and authorization information in the XML file. (Wassim Itani, &Ayman Kayssi, 2004:19)

The lack of XML support caused the XML security protocol unimplemented in MIDP due to the reason of the limitation of the string function provided by the java classes that reduce the efficiency of XML parsing. (Wassim Itani, &Ayman Kayssi, 2004:19)

2.1.2 Design Features for Secure Mobile Banking

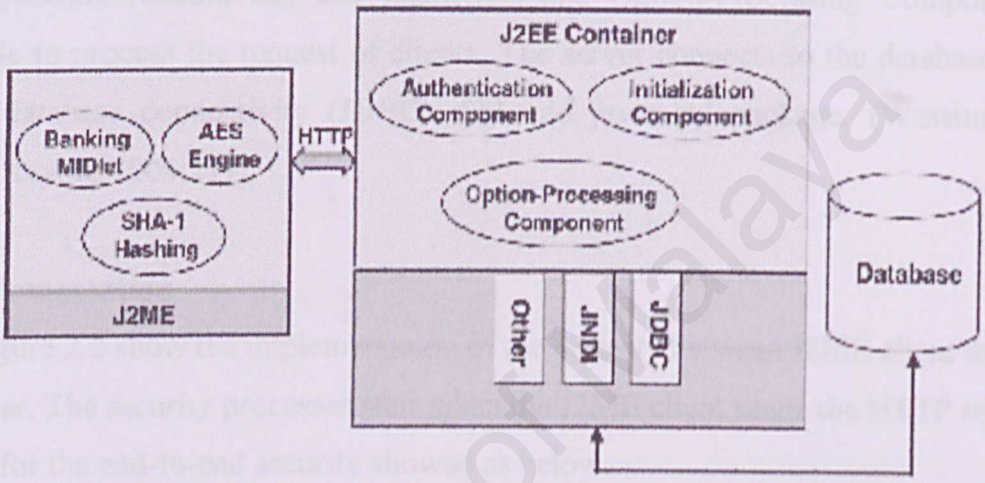


Figure 2.1: Client and Server components (Source: (Wassim Itani, &Ayman Kayssi, 2004:21, Obtained: September 26, 2005)

Figure 2.1 show the architecture of the Mobile Banking which consists of J2ME clients and J2EE Server via HTTP connection and Database connected to the server. (Wassim Itani, &Ayman Kayssi, 2004:21)

2.1.2.1 The client environments

The J2ME client application is developed using the MIDP 1.0 from J2ME wireless toolkit 1.0.4. The application is tested using Sony Ericsson P800 Java phone on GSM network using GPRS-enabled simcard. (Wassim Itani, &Ayman Kayssi, 2004:20)

The components in the J2ME clients included BankingMIDlet class which responsible for the main operation of the mobile banking system; AES Engine class which responsible for the encryption/decryption operation; and SHA-1 Hashing class which responsible for hashing operation. (Wassim Itani, &Ayman Kayssi, 2004:21)

All the class files in the J2ME application will be packaged into JAR and then downloaded the JAR file into the mobile phone together with Java application descriptor file. (Wassim Itani, &Ayman Kayssi, 2004:21)

2.1.2.3 The server environments

In the server side, there are three main components, include Authentication Component which is used to authenticating the client; Initialization Component which used to generate random key and challenge; and Option-Processing Component is responsible to process the request of clients. The server connects to the database using the Java database connectivity (JDBC) API and javax.sql package. (Wassim Itani, &Ayman Kayssi, 2004:21)

2.1.3 Implementation

Figure 2.2 show the implementation of the security between J2ME client and J2EE server. The security processes start when the J2ME client sends the HTTP request. The steps for the end-to-end security showed as below:

1) Generate the 128-bit random number (challenge)
   - The challenge generated is to prevent the eavesdropper get the same value whenever the client sends authentication data.
2) Create new HTTP session for client
   - The session is used to identify the client by server. The session will also hold the clients' data and shared among the components in server.
3) Store the challenge in the session variable
   - The challenge will be store in the client session for authentication process.
4) Generate the 128-bit encrypt/decrypt session key
   - The encrypt/decrypt key also generated randomly in server for different session
5) Generated session key stored at client record database.
   - The storage of session key actually is optional whether the key is store in client record database or session variable. However, storing the key in session variable will destroy the key at the end of session.

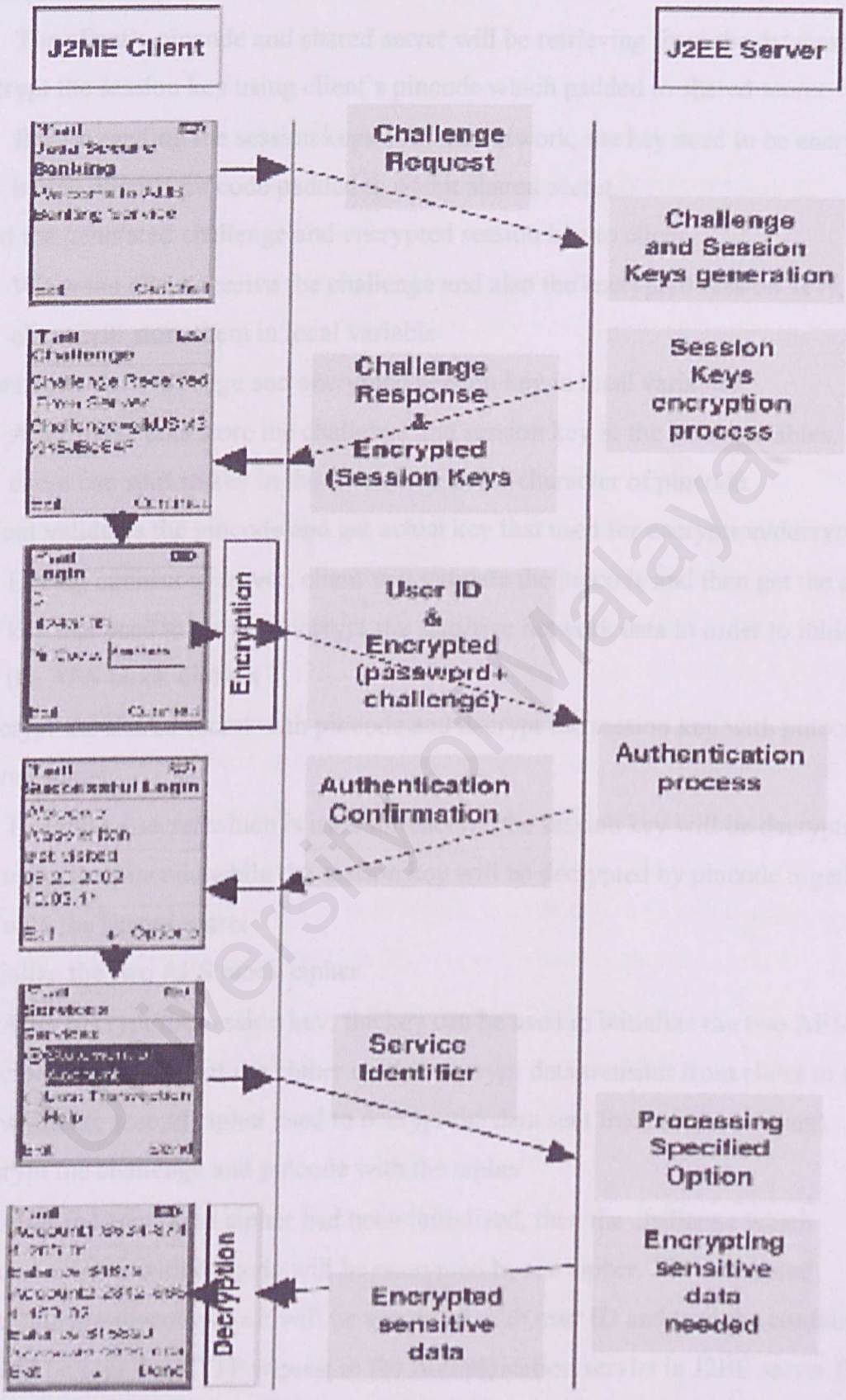


Figure 2.2: Secure client/server interactions (Source: Wassim Itani, &Ayman Kayssi, 2004:28, Obtained: September 26, 2005)

6) Retrieve client's pincode and shared secret from databases

- The client's pincode and shared secret will be retrieving from the databases.

7) Encrypt the session key using client's pincode which padded to shared secret

- Before sending the session keys over the network, the key need to be encrypted by the client's pincode padded to 64-bit shared secret.

8) Send the generated challenge and encrypted session key to client

- When the client receive the challenge and also the encrypted session keys, the client will store them in local variable

9) Client store the challenge and encrypted session key in local variables

- After the clients store the challenge and session key in the local variables, the client can start to key in the user name and 8 character of pincode.

10) Client validates the pincode and get actual key that used for encryption/decryption

- During connect to server, client will validate the pincode and then get the actual key that used to encrypt/decrypt the sensitive network data in order to initialize the AES block ciphers.

11) Decrypt the shared secret with pincode and decrypt the session key with pincode and shared secret

- The shared secret which is used for encrypt the session key will be decrypted using the pincode while the session key will be decrypted by pincode together with the shared secret.

12) Initialize the two AES block cipher

- After decrypt the session key, the key can be used to initialize the two AES block cipher which one of the cipher used to encrypt data transmit from client to server while the second cipher used to decrypt the data sent from server to client.

13) Encrypt the challenge and pincode with the cipher

- After the encrypted cipher had been initialized, then the challenge which concatenate with pincode will be encrypted by the cipher. The encrypted challenge-pincode block will be appended with user ID and then the combination will be sent via HTTP request to the Authentication servlet in J2EE server for authentication process. After the client being authenticated, they can start to use the applications. (Wassim Itani, &Ayman Kayssi, 2004:24-27)

2.2 SECURITY RISKS

2.2.1 Software Risk

As mention previously, the switching protocol process tends to decrypt data at the WAP gateway from WTLS and re-encrypt the data again before sending to SSL protocol. For this situation, if the hackers able to compromise the WAP gateway, the data can be easily derived during the decryption of data before it re-encrypt. (Anup K. Ghosh, & Tara M. Swaminatha, 2001:53)

2.2.2 Platform Risk

The operating system of the mobile device is a fundamental part to run the implementation of application. Without the security apply on the device, it is difficult to secure the data of application. Many manufactures tend to not include the following components into device, such as memory protection for process; file access control; authentication of principals to resources; differentiated user and process privileges; or biometric authentication. (Anup K. Ghosh, & Tara M. Swaminatha, 2001:54)

For examples, without the protection of memory, the legal application that sign the document using the private key will be attack by another application that tend to steal the decrypted key to signing the memory. The device also required to enforce the access control to prevent the unauthorized programs or user to access the sensitive information. It is better that the wireless device can be plug-in the biometric mechanism to provide stronger authentication such as the fingerprint recognition. Other than that, the software certificates shall be used to authenticate the software to user before installs them on the device. (Anup K. Ghosh, & Tara M. Swaminatha, 2001:54)

2.2.3 Software Application Risk

Software is relatively as important as platform run on device. However, the limitations on device such as memory and power limitation force the developer to forgo the security of software to improve the performance. (Anup K. Ghosh, & Tara M. Swaminatha, 2001:54)

2.3 CHAT ROOM.

2.3.1 Chat room system

2.3.1.1 Comparing Chat and Spoken Conversation

Chat is become popular at this age of millennium years. However, compared to the spoken interaction, chat has a few limitations such as poor interruptions managements, poor in organizing turn-taking, and poor in comprehension. (Smith, JJ Cadiz, & Byron Burkhalter, 2000:97)

The Conversation Analysis (CA) which is a sociological study of structure of ordinary face-to-face and spoken interaction had cited out that people using the fine tune and ordinary techniques can maintain the understandable spoken conversations. During spoken conversations carried out, there will be an organized turn and response structure that govern the group of people on how to exchange turns of talk. In the chat, the turn is organized according to the arrival of message in the central server and this will incline in the confusion of short message especially for those messages with ambiguity meaning. (Smith, JJ Cadiz, & Byron Burkhalter, 2000:97)

2.3.1.2 Problems with Text Chat

There are five core problems with the text chat which are:

- Lack of links among peoples and understandable of what they say
  - The exchange messages in the chat room among many peoples make difficulties in differentiate who is the speaker. Most of the chat systems address this problem by using the color or font to represent a particular of speakers (Smith, JJ Cadiz, & Byron Burkhalter, 2000:98)
- No visibility of listening-in-progress
  - In the chat, people will not be able to express their reaction when listening; this will sometimes lead to misunderstood and lose the senses of social presence. Erickson et al.'s Babble had addressed this issue by using a graphic design representing the activity of participants. This will create the intuitive sense of recently active people

  (Smith, JJ Cadiz, & Byron Burkhalter, 2000:98)

- Lack of visibility of turns-in-progress
  - In the chat system, the turns only transmit when user hit the ENTER key. This mean that producing the message is separate from transmitting the message. This will cause the chat system not synchronous. If there is a delay in the chat system whereby the user may type the text very slowly or leave the chat room for a moment, other people will misinterpret that person is not interesting on the topics discussed. Microsoft MSN addressed this issue by displaying the message "[Name] is typing the message" when the user is typing the message. This will indicate other people that person is going to transmit the message.

  (Smith, JJ Cadiz, & Byron Burkhalter, 2000:98)

- Lack of control in turn positioning
  - In the chat system, there is lack of turn positioning as show in the chat interaction below:

```
1 Larry: boy do we need to work on our
         interview skills....
2 James: who's conducting the
         interviews, anyway?
3 Scott: Yes
4 James: okay...
5 Larry: All of us
```

  The interactions show that the following message such as "All of us" do not fit to the prior turn which is "okay". The only way to do is to scroll up to find the candidate's prior turn. Babble suggests that slower interaction rate can increase people certainty in their position of turn.

  (Smith, JJ Cadiz, & Byron Burkhalter, 2000:99)

- Lack of social context and useful recordings
  - Chat room system normally doesn't have social history. As the visibility had scroll out of every user's history buffer, the content will disappeared.

(Smith, JJ Cadiz, & Byron Burkhalter, 2000:99)

2.3.2 Threaded Chat

Threaded chat is a structure chat room that solve the problems about the lack of social history and unorganized of turn sequences. The user interface of Threaded chat is shown in the Figure 2.3.

Threaded chat organizes the turns into structure tree whereby the turns are directly connected to the turn that they are going to response and it provide a facilities to drag the turns into correct location if the turns are misplaced. The thread (response structure) can be growing in unlimited size.

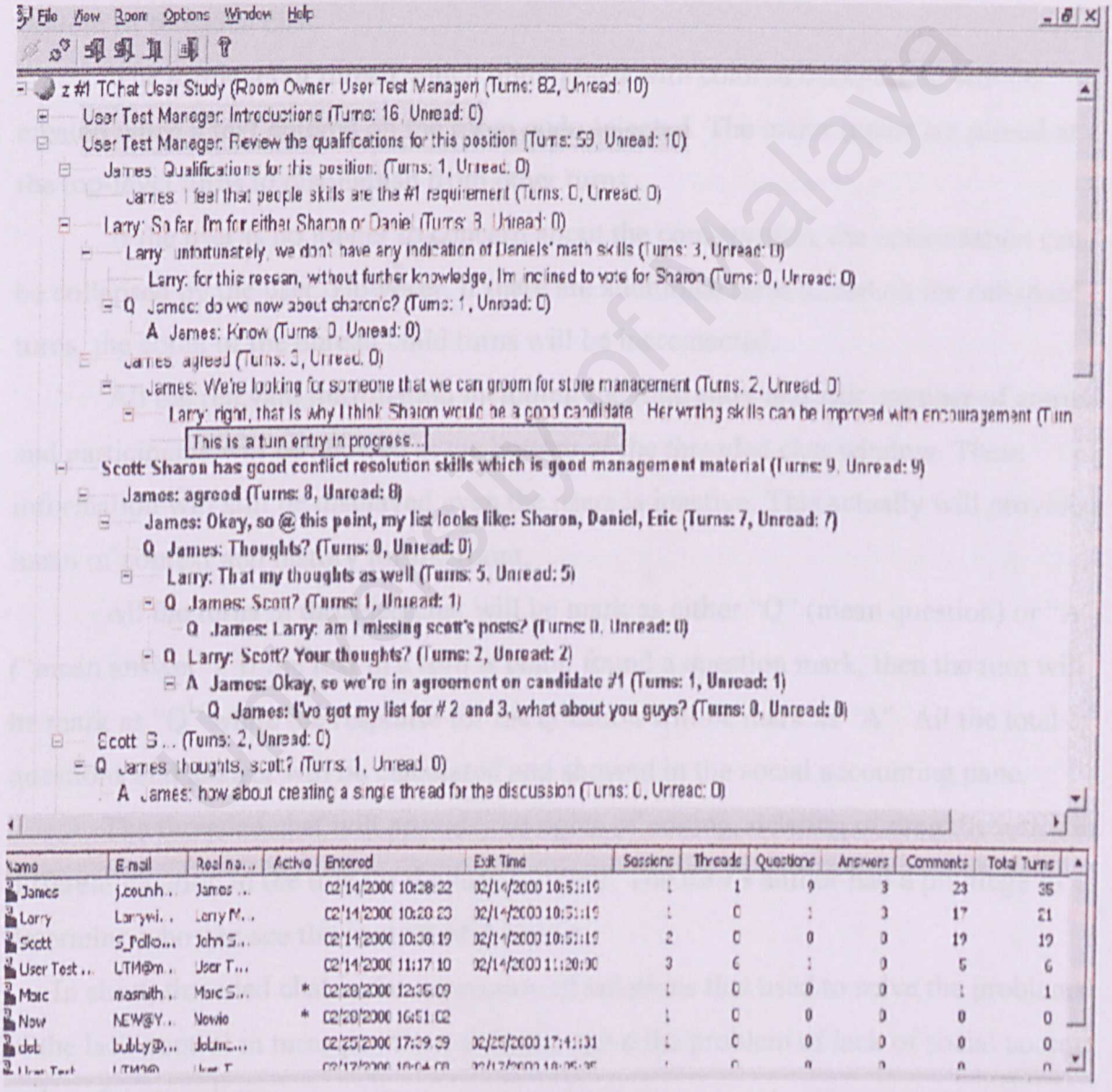


Figure 2.3: The Threaded Chat user interface (Source: Smith, JJ Cadiz, & Byron Burkhalter, 2000:98, Obtained: August 28, 2005)

When the users want to chat, they just need to click at the turn that they want to response to and type the message at that location. During the message typing, the message ("Entering Text") will be displayed to all other users. After finish typing the message, the users just need to hit the return key in order to make the message to visible to everyone.

The significant of the threaded chat is that the new message will be displayed in bold font and the font will fades to gray over the time to ease people to attract to the latest message. The turns will become unbolded when the users click on the turn to read or reply. There will be a count to display the numbers of unread turns and reply at the bottom of threaded chat.

The top-level of thread which highlighted with colored background will be created when a text entered on the room node selected. The major topics are placed at the top-level turns to distinguish from other turns.

If the user is no longer to concern about the conversation, the conversation can be collapsed by the user. However, if there are additional turns added on the collapsed turns, the count of the unread child turns will be incremented.

All the relevant information including the time entry and exit, number of entries and participants will be showed at the bottom of the threaded chat window. These information will still be displayed even the users is inactive. This actually will provide a mean of context and history for the room.

All the turns in threaded chat will be mark as either "Q" (mean question) or "A" ("mean answer"). If the text in a turn is being found a question mark, then the turn will be mark as "Q" while the response for the question will be mark as "A". All the total of questions and answer will be calculated and showed in the social accounting pane.

The threaded chat will provide the rights of editing, deleting or drag the turn s in different location in the tree for the turn's author. The turn's author had a privilege to determine who can see the content of the turns.

In short, threaded chat is the suggestion of solutions that used to solve the problems of the lack control in turns positions and also solve the problem of lack of social context and useful recordings

(Smith, JJ Cadiz, & Byron Burkhalter, 2000:100-101)

2.4 ANALYSIS OF REVIEW

The chat room module in the Mobile UM Map required the users to register first in order to get the valid user name and password to login into the system. The purpose of the login is to authenticate and identify the user when they use the chat room. Thus, it is important for the Mobile UM Map to provide the security solutions in order to prevent unauthorized users.

The first section of the review introduces the security solutions in the transport layer and application layer. After that, the review also show the process of encryption that the security solutions implemented in Mobile Commerce. In the server, a random challenge will be generated when the J2ME clients send the request and a HTTP session will established that will used to identify the users by the server. These concepts can be implemented in the Mobile UM Map as when the users of Mobile UM Map login, the session also will be generated for particular clients. Thus, the server can identify who is the user when the multi users using the chat room system in the Mobile UM Map.

However, the security solutions that adopt the encryption suggested by Wassim Itani, and Ayman Kayssi are not really secure as it can be decrypted. So, I intend to use hashing to hash the user passwords as the hashing technique will not allow the thing, so called "dehash". The hashing technique will help to hash the password into the message digest. The message digest is actually used to package the large piece of data into a fix length of digest value. So when there is eavesdropper trying to steal the data, they just only can get the digest value, not the whole data being sent.

The mobile application required some of the data security needs, such as integrity, authentication, and confidentiality. The cryptography will surely provide some solutions to meet the security needs. However, the Java Cryptography Architecture (JCA) and Java Cryptography Extension (JCE) that provide by Sun Microsystem cannot be supported in J2ME. Thus, it is being suggested to use Bouncy Castle cryptography package which is the open source effort in Australia.

In the chat room module, Message digest will be used to hash the password into cipher text at the server and then store in the MySQL database. The hashing technique can be done by using the MD5. The following shows the MD5 codes:

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

/**
 * A pretty speedy way to do MD5 hashes.
 */
public abstract class MD5 {

	// we don't really need more than one instance.
	private static MessageDigest algorithm;

	static {
		try {
			algorithm = MessageDigest.getInstance("MD5");
		}

		catch (NoSuchAlgorithmException e) {
			; // NoSuchAlgorithmException... it's friggen MD5.
		}
	}

	// for generating the String.
	private static char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7',
			'8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

	/*
	 * As I'm not smart enough to write my own MD5 hashing algo, I've instead
	 * just opted to use the one that's now standardly shipped. The good news is
	 * that I did in fact write the String-generating bit, because I thought it
	 * was silly to use a StringBuffer for something like that.
	 */
```

```
    /**
     * Returns a 32 character String representing the hexidecimal 16-byte MD5
     * hash of the passed data.
     * @throws JamudException
     *         if the MD5 MessageDigest was unavailable.
     */
    public static String hash(byte[] data) throws NoSuchAlgorithmException {
        if (algorithm == null) {
            throw new NoSuchAlgorithmException("MD5 not available");
        }

        byte[] digest;

        synchronized (algorithm) {
            algorithm.reset();
            algorithm.update(data);
            digest = algorithm.digest();
        }

        // you'd be surprised just how much faster than a StringBuffer and a
        // bunch of Integer.toHexString() calls this is.
        char[] out = new char[32];

        for (int i = 0, j = 0; i < 16; i++) {
            out[j++] = digits[(0xF0 & digest[i]) >> 4];
            out[j++] = digits[0x0F & digest[i]];
        }

        return new String(out);
    }
}
```

When the user register, their password will be sent to the server and the password will be hashed before stored into MySQL database. The following show part of the code that uses the MD5 to hash the password:

```
if (!checkSet.next()) {
        String hashedPassword = MD5.hash(userPassword.getBytes());
calling MD5 to hash the password
        String insertQuery = "INSERT INTO User (userID, userPassword) VALUES('"
+ userID + "', '" + hashedPassword + "')";
        statement.execute(insertQuery);
        connection.close();
        return "500 REGISTERED";
}
```

After the user had registered as the valid member, they can login to the system. Their username and password will be sent to the server. In the server, the password that sent will be hashed and the hashed password that matches the username in the MySQL database will be compared. If the hashed password that sent is match with the hashed password in the MySQL database, then the user is allowed to enter the chat room system. The following shows the part of the codes that implement the user validation:

```
try {
        Class.forName(driver);
        Connection connection = DriverManager.getConnection(url, username,
password);

        Statement statement = connection.createStatement();
        String validateQuery = "SELECT userPassword, userStatus FROM mum.User
WHERE userID = '" + userID + "'";

        ResultSet validateSet = statement.executeQuery(validateQuery);
        validateSet.next();

        if(validateSet.getString("userStatus").equals("online")){
```

```
            return "501 HELD_BY_SOMEONE";
        }

        if
(validateSet.getString("userPassword").equals(MD5.hash(userPassword.getBytes()
))) compare the hashed password that sent with the hashed password in the database
{
        String updateQuery = "UPDATE User SET userStatus = 'online' WHERE userID
= '" + userID + "'";
        statement.execute(updateQuery);
        connection.close();
        return "200 AUTHORIZED";
        }
        connection.close();
        return "400 UNAUTHORIZED";
}
```

The last session is the issue regarding the chat room system. It is important to investigate some of the problem raise during using the chat room system. The problems include the unorganized of the turns and cause the confusing in the message delivery, difficulty in identify the people who post the message, and lack of listening progress that will caused the misunderstood of meaning.

Some of the solution regarding the problems including using different of text color to represent different users, display a message that show a person is typing a message and going to transmit the message, use graphical design to represent the activity of participants and design slow interaction rate, and to increase the certainty of participant in the position of turns.

If we compare the mobile phone chat room with the computer chat room, we will realize some of the differences between both of the chat room system as shown in the Table 2.1.

Table 2.1: Comparison between Mobile Chat room and Computer Chat room

Criteria	Mobile Chat Room	Computer Chat Room
1.Screen Size	Small	Can be adjust the size screen
2. Graphical support	Less graphical support	More graphical support
3. Ease of visibility	Poorer visibility as the screen is too small	Better visibility as the screen is big
4. Life time	Will be end when there is interruption	Can be open all the days as long as the network is connected
5. Function	Less function can be support	More function can be support
6.Others	Provide mobility way to use the chat room	Must sit in front of the computer to use the chat room

3.0 EXISTING SYSTEM

3.1 MOBILE CHAT ROOM

3.1.1 Viztel's BuddyTalk

Figure 3.1: Architecture of Viztel's BuddyTalk

(Source:http://www.viztel.com/products_buddytalk.htm, obtained August 24, 2005)

The Figure 3.1 is the architecture show the Viztel's BuddyTalk. The Viztel's BuddyTalk is a voice chat-room application and added the capabilities of Web-Profiling and Web Control. (ViztelSolutions, 2005)

The LifeBuddy TM Engine is used to interconnect the web and mobile functionalities so that the users are allow viewing the status of chat user via online webpage as shown in Figure 3.2. (ViztelSolutions, 2005)

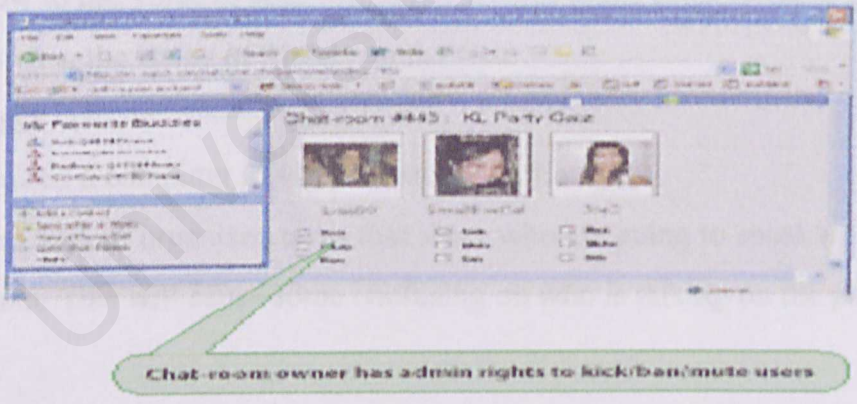


Figure 3.2: The web that show the status of chat users

(Source:http://www.viztel.com/products_buddytalk.htm, obtained August 24, 2005)

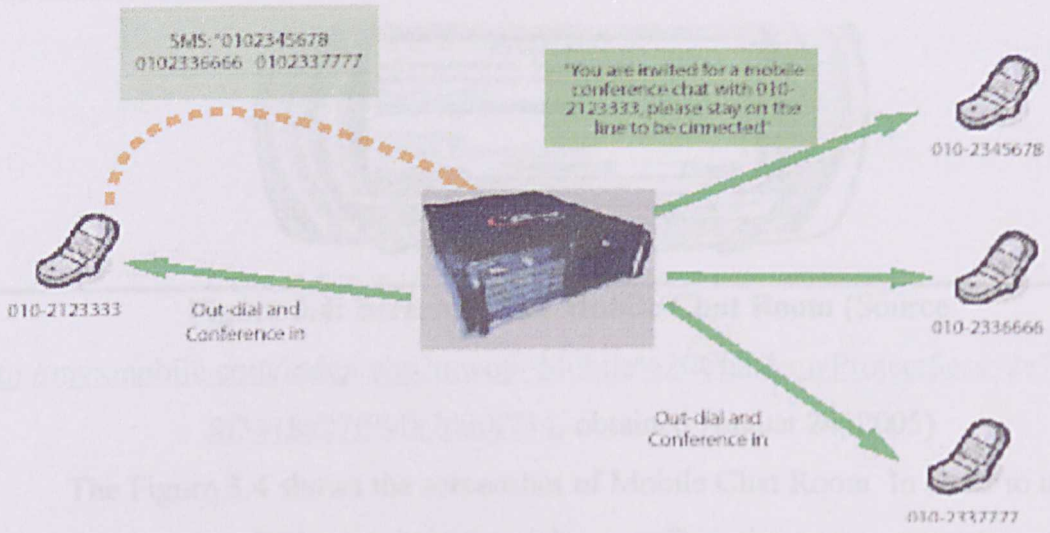


Figure 3.3: The process of inviting chat friends

(Source:http://www.viztel.com/products_buddytalk.htm, obtained August 24, 2005)

The Figure 3.3 shows how a user can invite his/her friends to join the buddy chat using SMS by out dial the invited friends' hand phone numbers. (ViztelSolutions, 2005)

Advantages

- Allow to use voice to chat with many friends at the same times in mobility way.
- Can view the status of friends on the web.

Disadvantages

- Required a computer to view the status of friend.
- There will no organized turns that show who are going to speak up through the system. This will cause some confusing on who is talking on the phone.

3.1.2 Mobile Chat

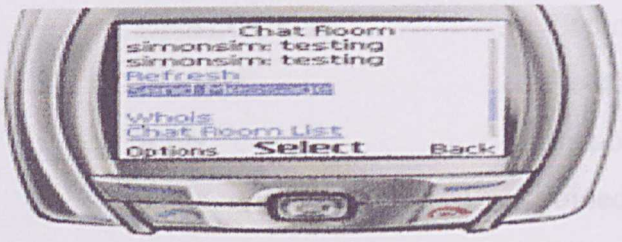


Figure 3.4: Screenshot of Mobile Chat Room (Source: http://myxmobile.com/index.php?mwop=Mobile%20Chat&myProjectSess=2e7590843a8f3418a27f99f87dd0f734, obtained August 24, 2005)

The Figure 3.4 shows the screenshot of Mobile Chat Room. In order to use the service, the user required to register the nickname. Then the user can start to send message to individual friend or send message to the chat room. Table 3.1 shows the list of instruction of the services. (myXMobile, 2004)

Table 3.1: Table show the instruction of services

KEYWORD	EXAMPLE	DESCRIPTION
NICK NAME (max.8 char & alpha numeric)	NICK ERRA	Register NICKNAME
MSG NICKNAME MESSAGE	MSG ERRA how r u?	Send personal message
BM	BM	Set all instructions to Bahasa Melayu
EN	EN	Set all instruction to English
SEARCH AgeGender	SEARCH 21F	Find new friend
WHOIS NICKNAME	WHOIS ERRA	Get friend's details
BLOCK	BLOCK	Get all the blocked name
BLOCK NICKNAME	BLOCK ERRA	Block all the blocked name
BLOCK ALL	BLOCK ALL	Block all the message
UNBLOCK NICKNAME	UNBLOCK ERRA	Unblock message from

		specific NICKNAME
UNBLOCK ALL	UNBLOCK ALL	Unblock all messages
ROOMS	ROOMS	Room lists
JOIN	JOIN	List of ROOMS that can be joined
JOIN ROOM	JOIN REMAJA	Join specific room
USERS	USERS	Users lists in the ROOM
CHAT	CHAT	Commands list in chat service
CHAT MESSAGE	CHAT hi all	Send message to all chatters in the room
LEAVE	LEAVE	Leave the chat room

(Source: http://myxmobile.com/index.php?mwop=Mobile%20Chat&myProjectSess=2e7590843a8f3418a27f99f87dd0f734, obtained August 24, 2005)

Advantages

- Easy to use as it just only type certain keywords for particular transaction.
- Allow to view the message that displayed earlier of discussion. However, users also can refresh the screen if they wish to.
- Allow to send personal message.

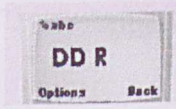
Disadvantages

- It is uncomfortable to type the keywords every time need to perform certain transaction.
- Need to remember the keyword when the user wants to perform the particular transaction.

3.1.3 DD-Chat

In order to use the service in DD-Chat, users need to follow the steps:

1)

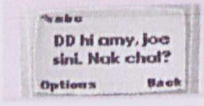


DD-Chat required users to register first by key in the DD R and then send to 32300 (ddMobile, 2003)

2)

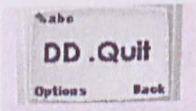
To join the chat room, the user needs to key in DD.join<RoomId> and send to 32300. If want to change the room, the user required to type the keywords again. (ddMobile, 2003)

3)



To send message, the user needs to type DD<type message here> and send to 32300. (ddMobile, 2003)

4)



To quit, the user needs to type DD.Quit and send to 32300. (ddMobile, 2003)

5)

To get all code, type DD.Help and send to 32300. (ddMobile, 2003)

Advantage

- Easy to use.

Disadvantages

- The limited of functions.
- Inconvenience to users as they need to type keywords and SMS to 32300 every times they want to perform a particular transactions.

3.1.4 Mocha

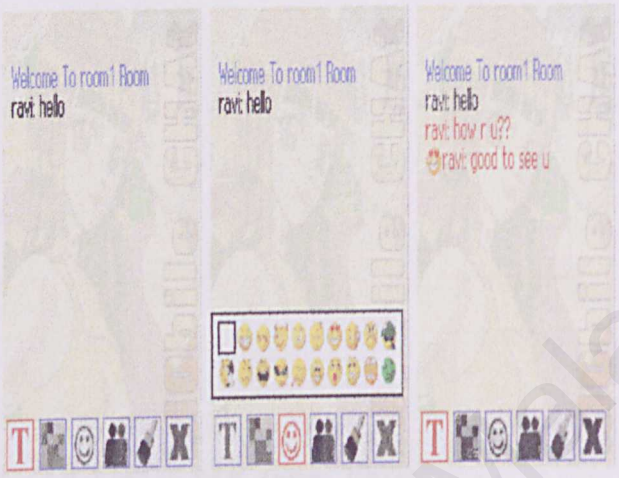


Figure 3.5: MoCha Screenshot (Source: http://www.zgroup-mobile.com/published_games/application/mocha/mocha.html, Obtained August 18, 2005)

The Figure 3.5 is the Screenshot of one of the Mobile Chat Room called Mocha. The usefulness of the icons shown in the screenshot is as below:

1) ICON 1 – Type message
2) ICON 2 – change font color
3) ICON 3 – choose emotion
4) ICON 4 – send private message to particular friends
5) ICON 5 – Clear screen
6) ICON 6 – Exit and back to Home (Mocha)

Advantages

- provide the icons for users to choose
- attractive interface

Disadvantages

- The symbols in the icons are not understandable.

3.1.5 M-Chat

M-Chat allow users to carry out the chat rooms publicly with many others people even the person not in the friend list. The following show the steps in implement the M-Chat. (mtc, 2003)

1) View Room List

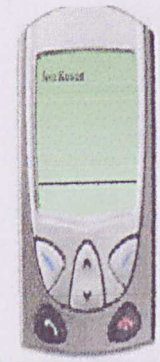
Send "chat" to 858. (mtc, 2003)

First MT Received

Second MT Received

Each chat room is given unique number. (mtc, 2003)

2) Join



Type Join & chat room name and send it to 858

Received SMS

Type "join" and chat room name and send it to 858

(mtc, 2003)

3) List of uers in chat room

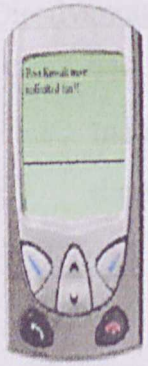
Type List & chat room name and send it to 858

Received SMS

Type "list" and chat room name and send it to 858

(mtc, 2003)

4) Send messages



Type Post, Name of Chat Room & your Message and send it to 858

OR

Type Post & message send it to 858xxxx *(Chat Room number)*

Type "post", name of chat room and message to 858

(mtc, 2003)

5) View message

All the login users of Chat Room "Kuwait" will receive message from Ahmed

All the users of the chat room will receive the message on SMS

(mtc, 2003)

6) Quit

Type Leave & chat room name and send it to 858

Received SMS

Type "Leave" & chat room name and send to 858

(mtc, 2003)

Advantage

- Easy to use.

Disadvantages

- The limited of functions.
- Inconvenience to users as they need to type keywords and SMS to 32300 every times they want to perform a particular transactions.

3.2 COMPARISON OF MOBILE CHAT ROOM

The first Mobile chat room is Viztel's BuddyTalk where the user required inviting the friends to join the chat room by SMS the friends' hand phone numbers. The status of the friends will be display in the web page on mobile device or computer. The Viztel's BuddyTalk is a voice chat application. The user can chat through mobile phone and each of the friends can get the same message in the same chat room.

The second Mobile chat room is Mobile chat. This mobile system will required user to type in the keywords to perform some operation. The same actions also need to perform on the third Mobile chat room that is DD-Chat. Both of them are quite troublesome because required users to key in the keywords every times need to perform a single operation. Moreover, the DD-Chat will overwrite the old message in chat room with new message. This is very undesirable design for users.

Another Mobile chat room is Mocha. This mobile system already prepares the icons to let users to choose to perform a particular operation. Apart from that, the interface of the chat room also attractive and colorful.

The final mobile chat room which is M-Chat also required the users to type the keywords and SMS to the particular parties. It also is facing the same problems as DD-Chat and Mobile chat.

In short, the best Mobile Chat room would be the Mocha because it provides a facility for user to choose to perform an operation. This is very comfortable for user compare the other which required user to type the keyword before perform the operation.

4.0 SUMMARY OF REVIEWS

In the literature review, I had research about the J2ME end-to-end security. After that, I introduce the MMAPI which is use to develop the J2ME audio and video and some issue about the chat room. Finally, I compare among the existing Mobile Chat Room.

As discuss previously, there is different security solutions reside in different layer which is transport layer and application layer. Each of the security solutions will have pros and cons. In the transport layer, the better security solution would be applying the KSSL in the J2ME device. This will allow the secure and direct communication with many web servers. It may affect the performance since the full handshake need to be accomplished each time whenever want to contact the new web server. This is unavoidable because as the nature of SSL, handshake need to be established before can communicate with server. In the application layer, it is suggested to use XML file to transmit data between client and server. However, the Sun Java Microsystems need to improve the XML as currently the XML is limited in the string functions and thus reduce the efficiency in XML parsing.

In the security implementation, the first step to contact server is to generate 128 bit random number (challenge) which can let the eavesdropper get different values when the client send authentication data for server. The second step would be created a HTTP session which used to identify the client. Then the next step is to stored challenge generated into the client's HTTP session for authentication use. After that, the 128-bit encryption/decryptions session key will be generated randomly by server. Then the key generated will be stored in the client's database record and will be used during authentication process. Before sending the session key to client over wireless link, the client's 64-bit pincode retrieved from database will padded to 64-bit shared secret (which is encrypted in the MIDlet's JAR file) to encrypt the session key. Then finally the session key together with the challenge can safely transmit to J2ME client and stored in the phone memory.

After the client receive the encrypted challenge and session key, both of them will be stored in local variables. Then the client can start to enter their user ID and 64-bit pincode. During connect to server, client will validate the pincode and then get the

actual key that used to encrypt/decrypt the sensitive network data in order to initialize the AES block ciphers.

The shared secret which is used for encrypt the session key will be decrypted using the pincode while the session key will be decrypted by pincode together with the shared secret. After decrypt the session key, the key can be used to initialize the two AES block cipher which one of the cipher used to encrypt data transmit from client to server while the second cipher used to decrypt the data sent from server to client.

After the encrypted cipher had been initialized, then the challenge which concatenate with pincode will be encrypted by the cipher. The encrypted challenge-pincode block will be appended with user ID and then the combination will be sent via HTTP request to the Authentication servlet in J2EE server for authentication process. After the client being authenticated, they can start to use the applications.

The security solutions discuss previously is using encryption. The cipher text encrypted can be decrypt by the people. Thus, there is less secure. Due to this reason, hashing is become a choice in the security of J2ME program as it cannot be "dehash". In the Mobile UM map, the digest message which generate through hashing will be used in implementing the security in the mobile system.

By using message digest, the user name and password together with the random number and timestamp will be calculated the digest value and send it to the server. The server will use the user name to look up the password belong to the user name and use the password to calculate the digest value in the server. The digest value created in the server that matches the digest value sent by the client will be authenticated as valid user.

After the authentication process, the clients can use the services provide by the application which is allowing the users to carry out the chat room among friends. Let review some issue about the chat room. There are a few problems faced in the chat system, including:

- Lack of links among peoples and understandable of what they say
- No visibility of listening-in-progress
- Lack of visibility of turns-in-progress
- Lack of control in turn positioning
- Lack of social context and useful recordings

The solutions to solve the problems stated, including:

- Use font or colors to represent a particular person
- Use graphical design to represent the activity of participants
- Display a message that show a person is typing a message and going to transmit the message
- Design a slow interaction rate to increase the certainty of participant in the position of turns

At the final part, I had compared the existing Mobile Chat Room. Most of the mobile phone chat system will require users to type a keyword in order to perform a particular operation. This will actually consume users' valuable times. It is better that Mobile Chat Room will prepare a set of icons represent a particular operation to let users choose in order to make the system more user friendly.

5.0 SYSTEM ANALYSIS

The chat room module has been analyzed to determine its functional and non-functional requirements. The following section shows the evidence of analysis process.

5.1 FUNCTIONAL REQUIREMENTS

The diagram 5.1 is the output of the analysis which is the use case that shows the functional requirements of chat room module in Mobile UM Map.

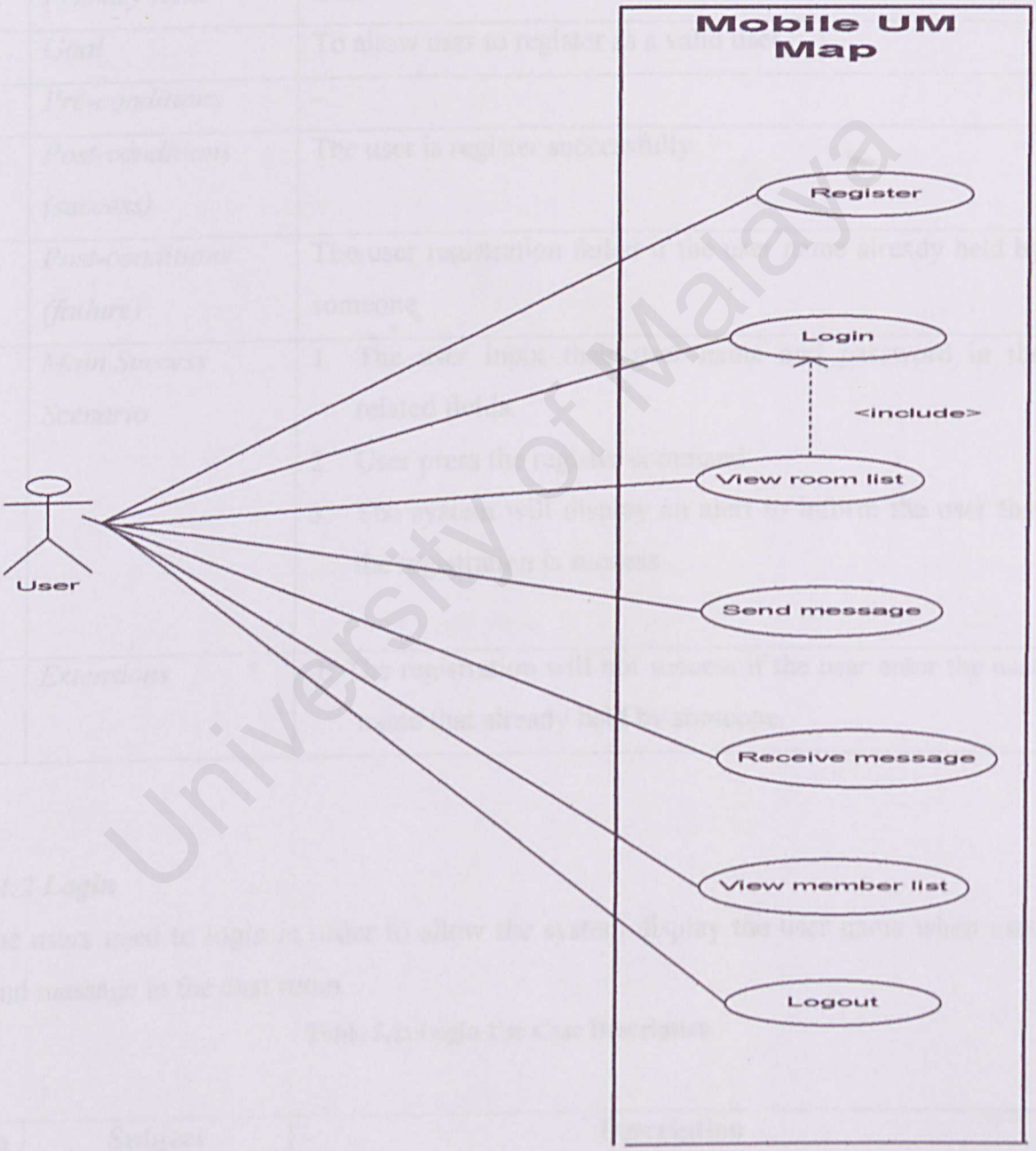


Diagram 5.1: Use Case of Chat room module

5.1.1 Register

The users need to register first before using the chat room module in the Mobile UM Map. They need to get the user name and password in order to login to the system.

Table 5.1: Register Use Case Description

No	Subject	Description
1.	*Use Case*	Register
2.	*Primary Actor*	User
3.	*Goal*	To allow user to register as a valid user
4.	*Pre-conditions*	-
5.	*Post-conditions (success)*	The user is register successfully
6.	*Post-conditions (failure)*	The user registration failed if the user name already held by someone
7.	*Main Success Scenario*	1. The user input their user name and password in the related fields. 2. User press the register command 3. The system will display an alert to inform the user that the registration is success
8.	*Extensions*	1. The registration will not success if the user enter the user name that already held by someone.

5.1.2 Login

The users need to login in order to allow the system display the user name when user send message in the chat room.

Table 5.2: Login Use Case Description

No	Subject	Description
1.	*Use Case*	Login

2.	*Primary Actor*	User
3.	*Goal*	To allow user login into the chat room module in the Mobile UM Map
4.	*Pre-conditions*	The user is a registered user
5.	*Post-conditions (success)*	The user is logon successfully
6.	*Post-conditions (failure)*	The user login failed
7.	*Main Success Scenario*	1. The user input their user name and password in the related fields. 2. User press the login command 3. The system will display the room list if the user login successfully
8.	*Extensions*	1. The login will not success if the user enter the wrong user name or password

5.1.3 View room list

The users can view a list of room together with the amount of members in each of the room. After that, the user can select the room they want in order to enter the selected room.

Table 5.3: View room list Use Case Description

No	Subject	Description
1.	*Use Case*	View room list
2.	*Primary Actor*	User
3.	*Goal*	To allow user view the room list
4.	*Pre-conditions*	The user login successfully
5.	*Post-conditions (success)*	The user able to enter the selected room
6.	*Post-conditions*	-

	(failure)	
7.	*Main Success Scenario*	1. The user view the room list 2. User select the room and press ok command 3. The system will display the selected room and the member amount
8.	*Extensions*	-

5.1.4 View member list

The users can view a list of members in the selected room by press the view member command. This will allow the user know who are in the room that they join.

Table 5.4: View member list Use Case Description

No	Subject	Description
1.	*Use Case*	View member list
2.	*Primary Actor*	User
3.	*Goal*	To allow user view the member list
4.	*Pre-conditions*	The user enter the selected room
5.	*Post-conditions (success)*	The user able to view the selected room
6.	*Post-conditions (failure)*	-
7.	*Main Success Scenario*	1. The user enter the selected room 2. User press the view member command 3. The system will display a list of members in the room
8.	*Extensions*	-

5.1.5 Send message

The users can send the message to the member in the room that the user chooses.

Table 5.5: Send message Use Case Description

No	Subject	Description
1.	*Use Case*	Send message
2.	*Primary Actor*	User
3.	*Goal*	To allow user to send message
4.	*Pre-conditions*	The user enter the selected room
5.	*Post-conditions (success)*	The user able to send message
6.	*Post-conditions (failure)*	Message unable to be sent
7.	*Main Success Scenario*	1. The user enter the message into the text box 2. User press the send command 3. The system will send the message to all the member that in the same room with the sender
8.	*Extensions*	-

5.1.6 Receive message

The users can receive the message when there have a member in the same room send the message.

Table 5.6: Receive message Use Case Description

No	Subject	Description
1.	*Use Case*	Receive message
2.	*Primary Actor*	User

3.	*Goal*	To allow user receive message
4.	*Pre-conditions*	The member in the same room send a message
5.	*Post-conditions (success)*	The user able to receive message
6.	*Post-conditions (failure)*	-
7.	*Main Success Scenario*	1. The user enter the room 2. The user receive message
8.	*Extensions*	-

5.1.7 Logout

The users can leave the room and logout in the room list. After that, the system will display the login form.

Table 5.7: Logout Use Case Description

No	Subject	Description
1.	*Use Case*	Logout
2.	*Primary Actor*	User
3.	*Goal*	To allow user leave the room
4.	*Pre-conditions*	The member login the system successfully
5.	*Post-conditions (success)*	The user able to leave the room
6.	*Post-conditions (failure)*	-
7.	*Main Success Scenario*	1. The user leave the room 2. The system display the room list 3. The user press the logout command and the system will display the login form
8.	*Extensions*	-

5. NON-FUNCTIONAL REQUIREMENTS

The chat room module also has fulfilled some of the non-functional requirements, such as maintainability, performance, security, reliability, and availability.

5.2.1 Maintainability

The chat room module has been developed using object-oriented method and each of the implementation are perform in separate function. In future maintenance, if anything modification is required on the particular implementation, only those function that affected by the implementation will be changed without affecting the other functions.

5.2.2 Performance

Through the testing using emulator, the messages that sent by the sender will be receive by the member in the same room within 5 seconds. Thus, the performance for the chat room module in the Mobile UM Map is high.

5.2.3 Security

The chat room module also has been considered the security issue. The chat room is required the user to registered as a valid member first. The password that they sent to the system will be hashed first before storing into the MySQL database. Thus, if there is any people successfully hack into the database, they can only view the hashed password, not the actual user password.

5.2.4 Reliability

The testing that has been carried out shown that the chat room module in the Mobile UM Map is in very high reliability because every 100 message that being sent, 100 message will be able to received by the members in the same room. Thus, the reliability of the chat room module reaches 100%.

5.2.5 Availability

The chat room module is available 24 hours per day if the server is online 24 hours. However, if the server is in maintenance process, the system will not be available.

5.3 HARDWARE AND SOFTWARE REQUIREMENTS

The Mobile UM Map is developed using Java language where the mobile client side is using the J2ME, server side using the Servlet, and the database using the MySQL database. The following show the tools that involved in developing the chat room module in Mobile UM Map.

5.3.1 Eclipse

Eclipse is an IDE that use to develop the Mobile UM Map. It is easy to use because it provide a capabilities to create the project easily and provide a lot of views to the developer to view some of the components required such as Console which display the output of the implementation, Package Explorer which show the structure of the project, Problem which list out the problems occurs after the codes are debugged, etc.

The Eclipse also can be set to install the Tomcat that can be start, stop, and restart within the Eclipse. It also can be set up to package the jar file of the mobile client system. Figure 5.1 shows the layout of the Eclipse.

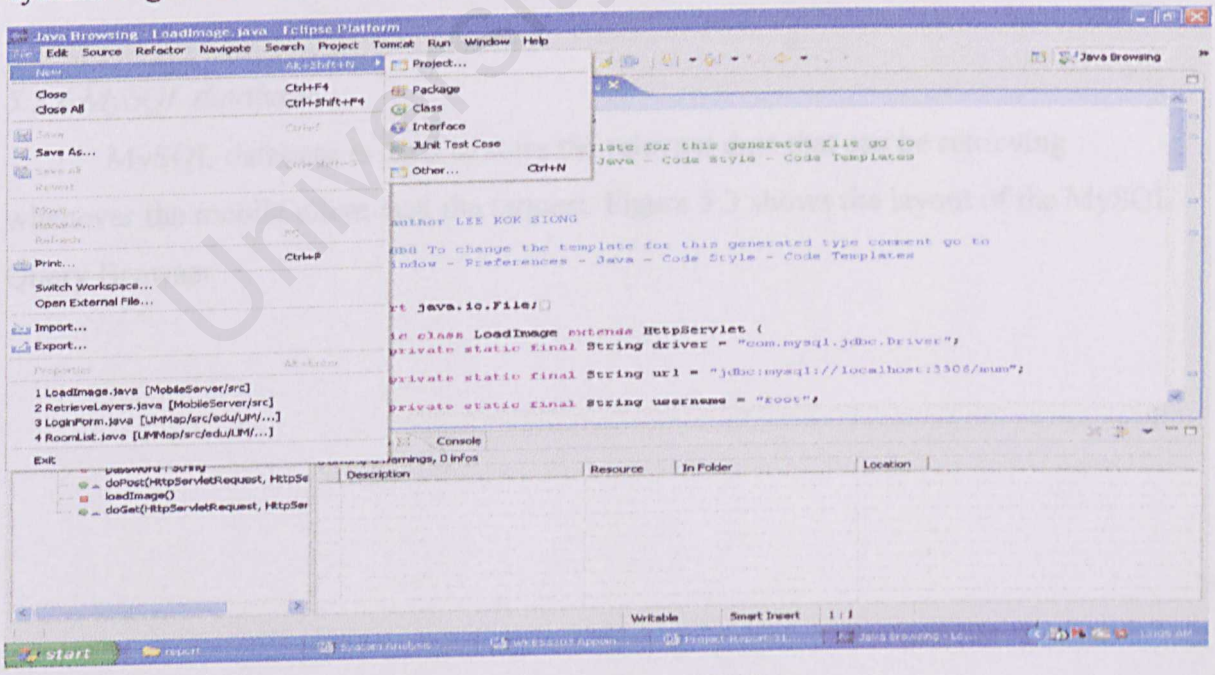


Figure 5.2: Layout of Eclipse

5.3.2 J2SDK

It is the essential components which need to be installing into the computer in order to implement the system using Java language.

5.3.3 J2ME Wireless Toolkits

It is the required component in developing the J2ME mobile client. It provides required API (MIDP 2.0) and the CLDC in developing the client side module.

5.1.3.4 Tomcat

Tomcat is the server chosen to deploy all the Servlet class for the Mobile UM Map into the platform (Tomcat itself). It can be connected by the mobile client using the http connection.

5.3.5 Nokia Emulator

Nokia series 60 provide a special feature which is the FullCanvas that can display the map in full screen of mobile phone. The Nokia Series 60 emulator act as a presentation layer which provide the interface for the users. It is substitute the real phone interface which allows the developer to do the testing easily when developing the mobile client module.

5.3.6 MySQL database

MySQL database is used to store the relevant data that can be retrieving whenever the mobile client sent the request. Figure 5.3 shows the layout of the MySQL Query Browser.

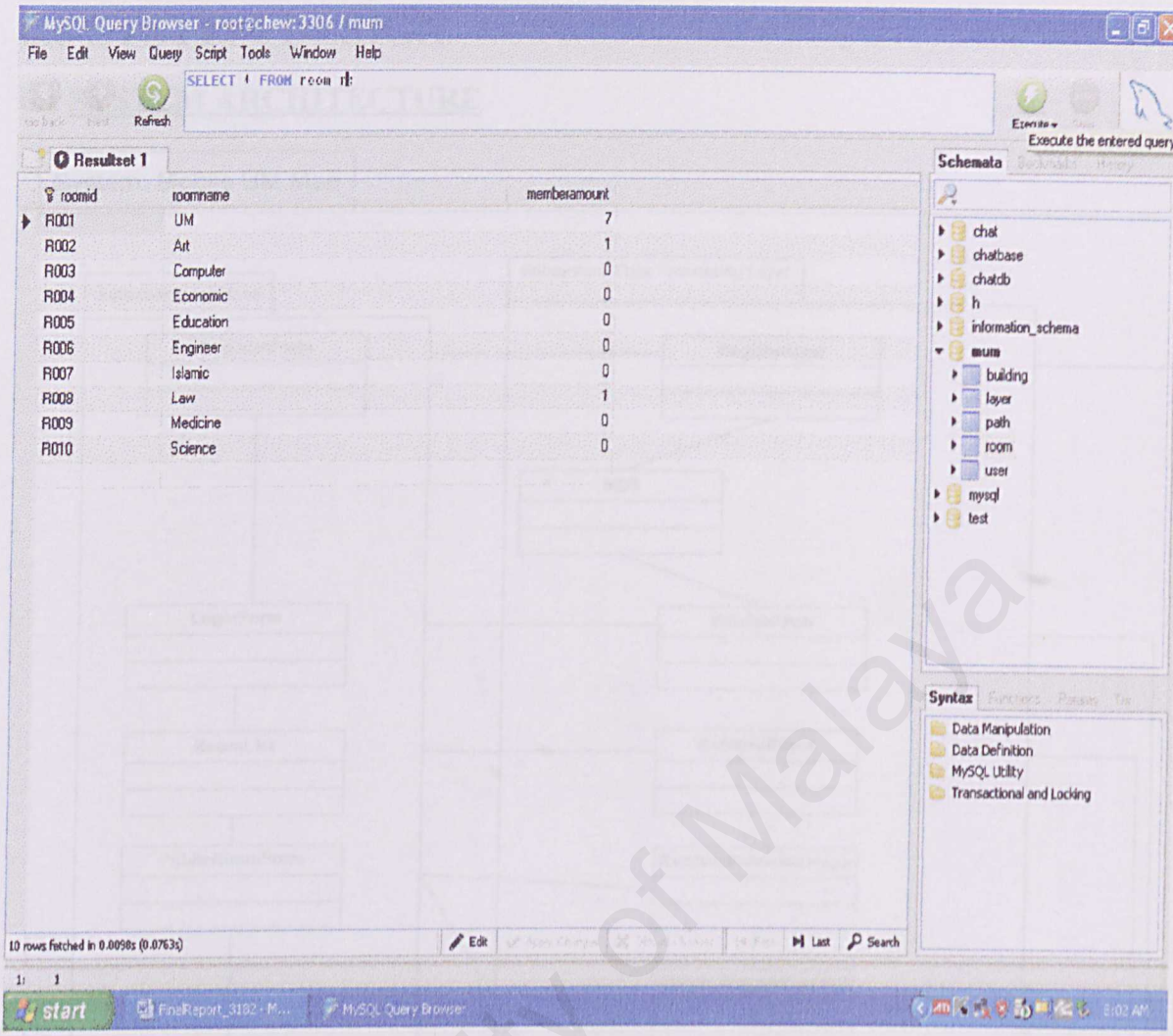


Figure 5.3: Layout of MySQL Query Browser

6.0 SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

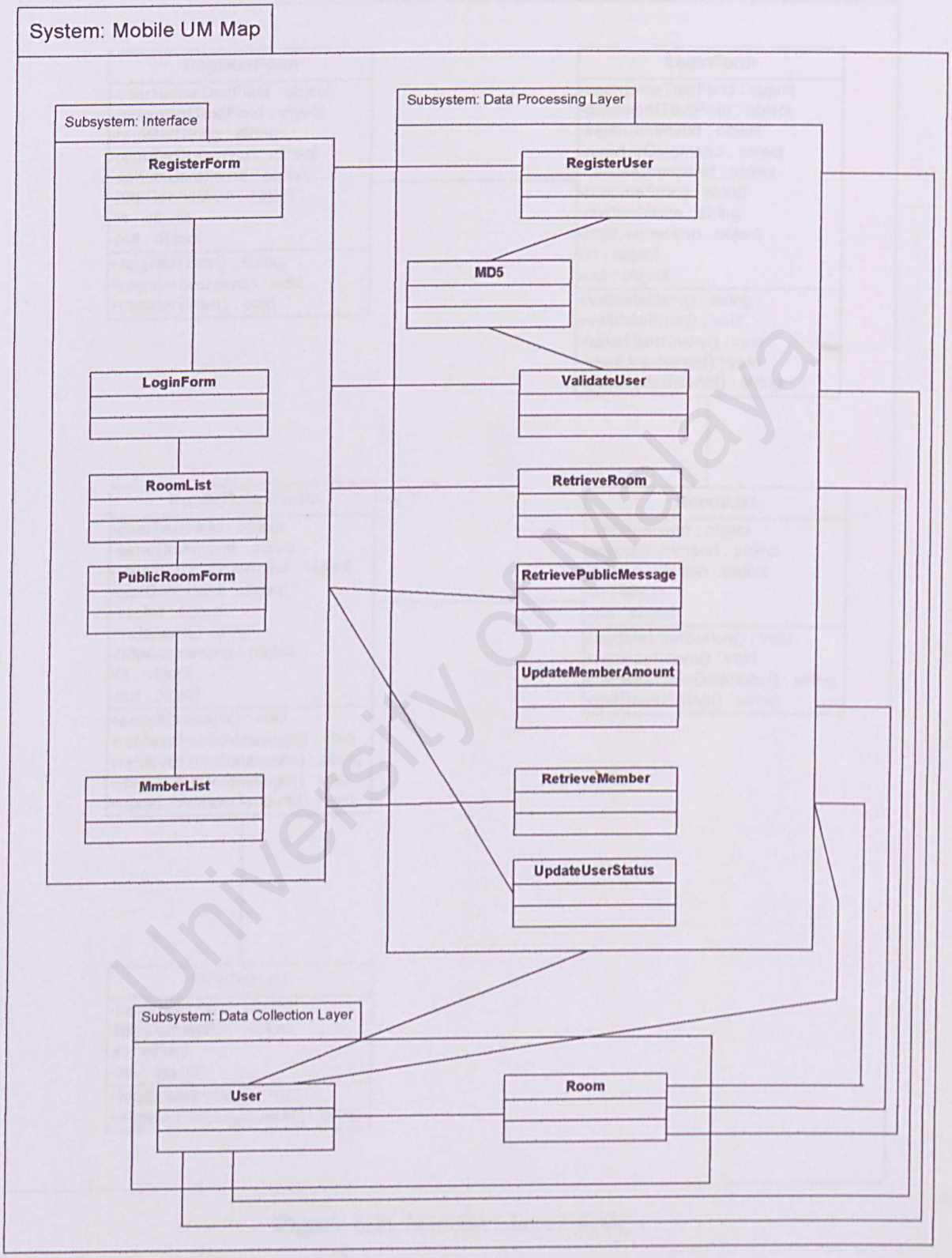


Figure 6.1: System architecture of Mobile UM Map

6.2. INTERFACE LAYER DESIGN

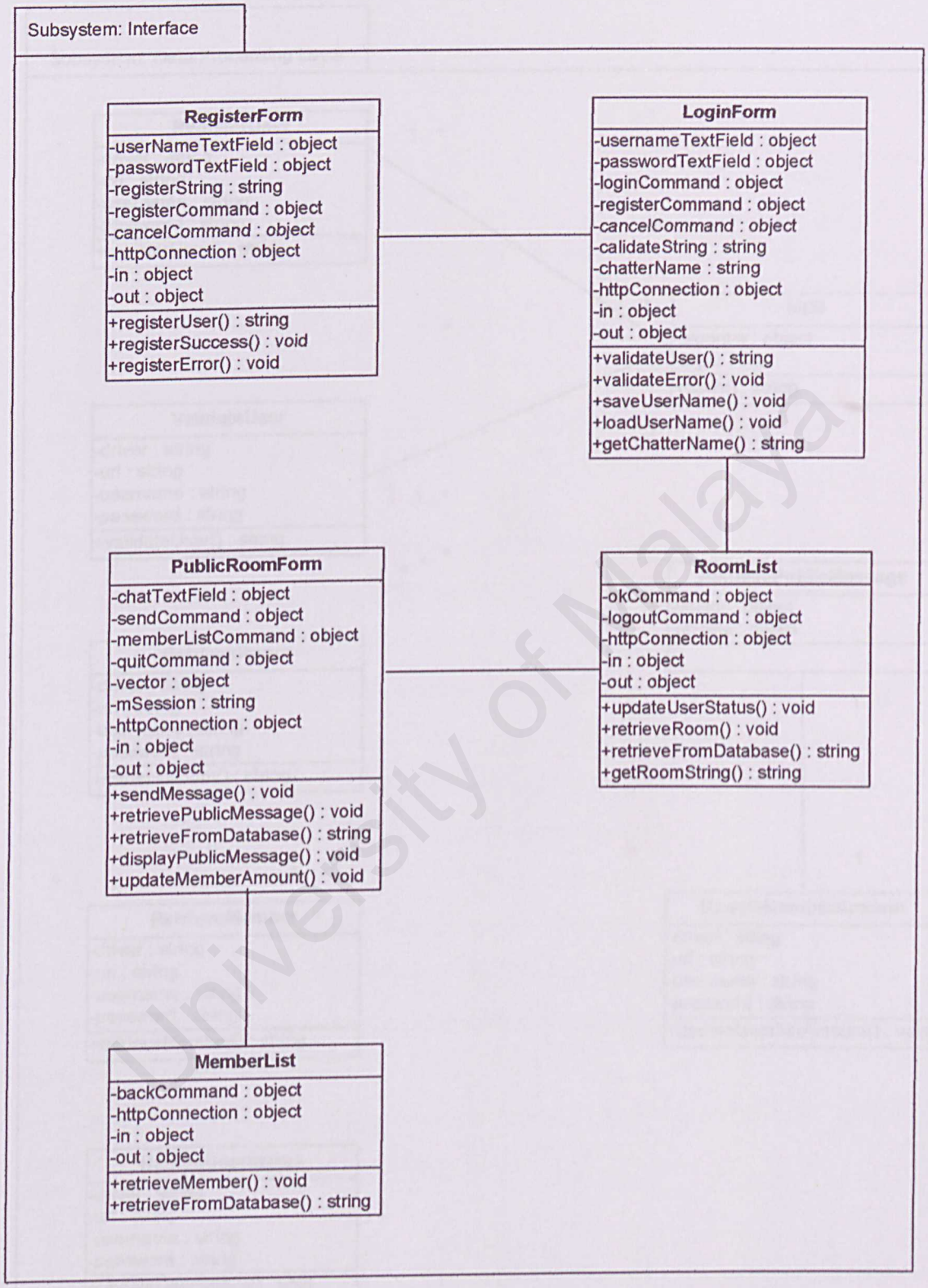


Figure 6.2: Interface layer design

6.3 DATA PROCESSING LAYER DESIGN

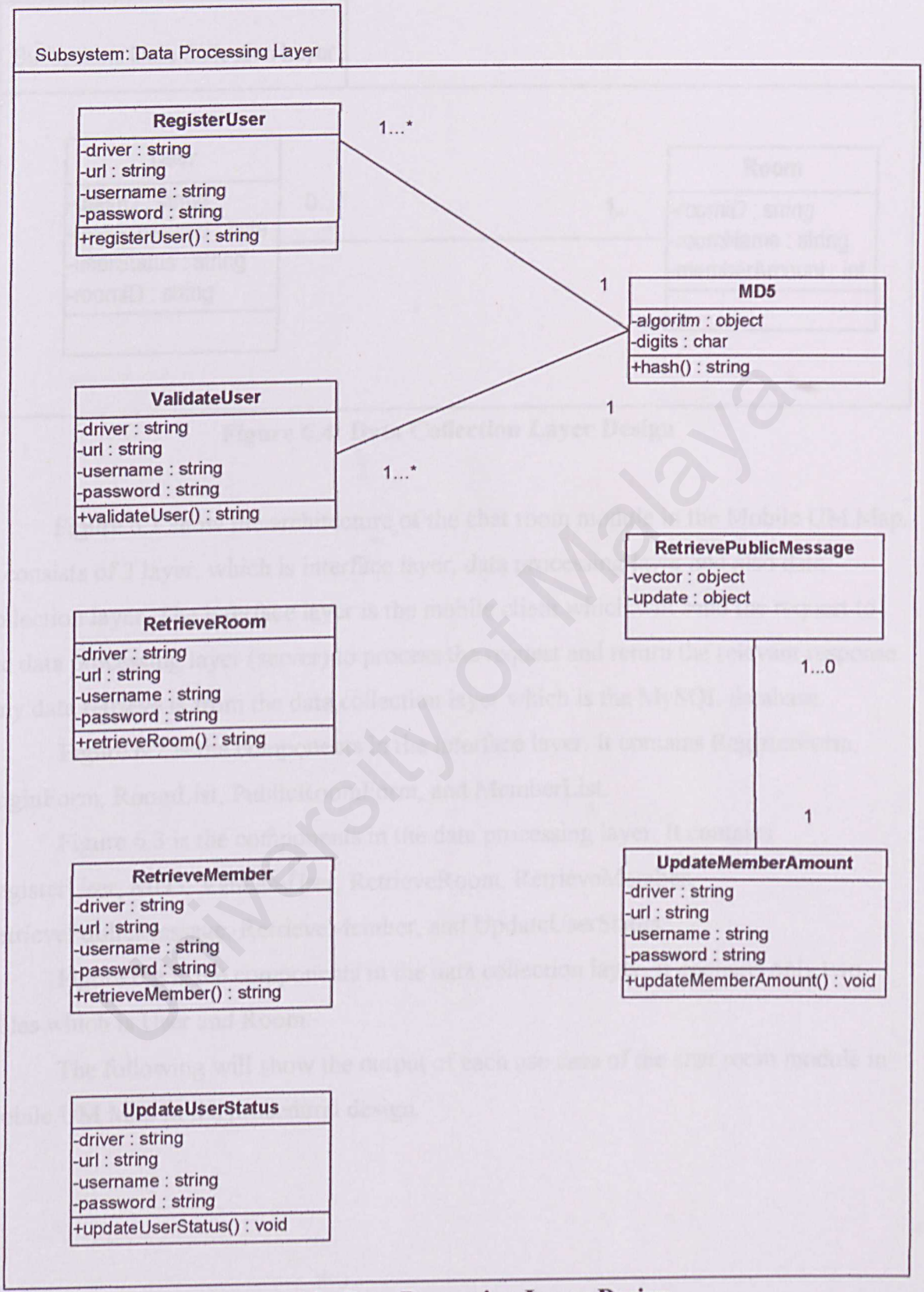


Figure 6.3: Data Processing Layer Design

6.4 DATA COLLECTION LAYER DESIGN

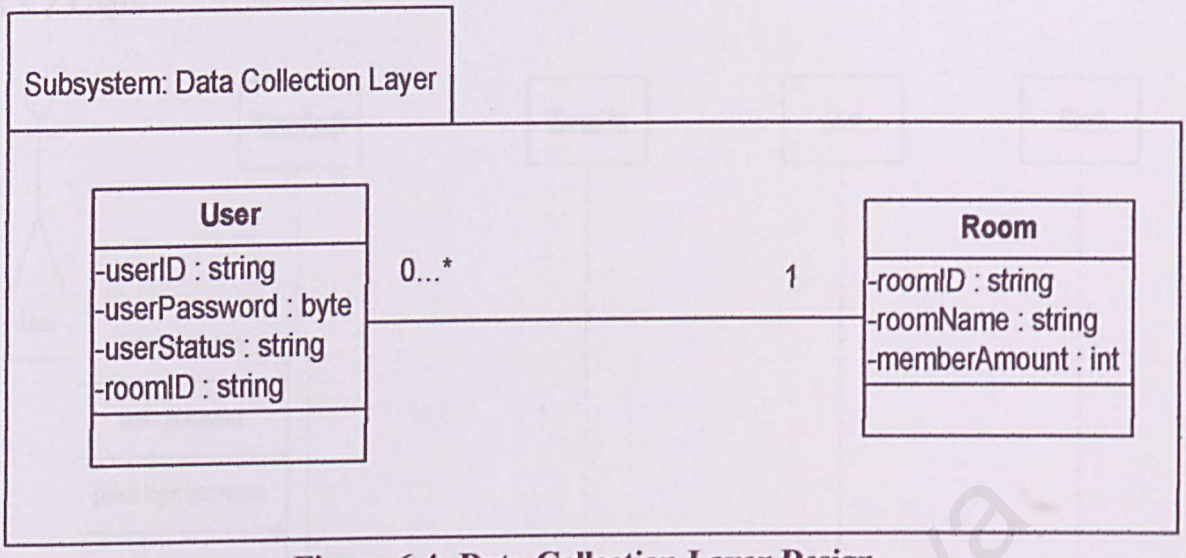


Figure 6.4: Data Collection Layer Design

Figure 6.1 show the architecture of the chat room module in the Mobile UM Map. It consists of 3 layer, which is interface layer, data processing layer and also data collection layer. The interface layer is the mobile client which will send the request to the data processing layer (server) to process the request and return the relevant response. Any data retrieve is from the data collection layer which is the MySQL database.

Figure 6.2 is the components in the interface layer. It contains RegisterForm, LoginForm, RoomList, PublicRoomForm, and MemberList.

Figure 6.3 is the components in the data processing layer. It contains RegisterUser, MD5, ValidateUser, RetrieveRoom, RetrieveMember, RetrievePublicMessage, RetrieveMember, and UpdateUserStatus.

Figure 6.4 is the components in the data collection layer. It contains only two tables which is User and Room.

The following will show the output of each use case of the chat room module in Mobile UM Map in the procedural design.

6.5 PROCEDURAL DESIGN

6.5.1 Login

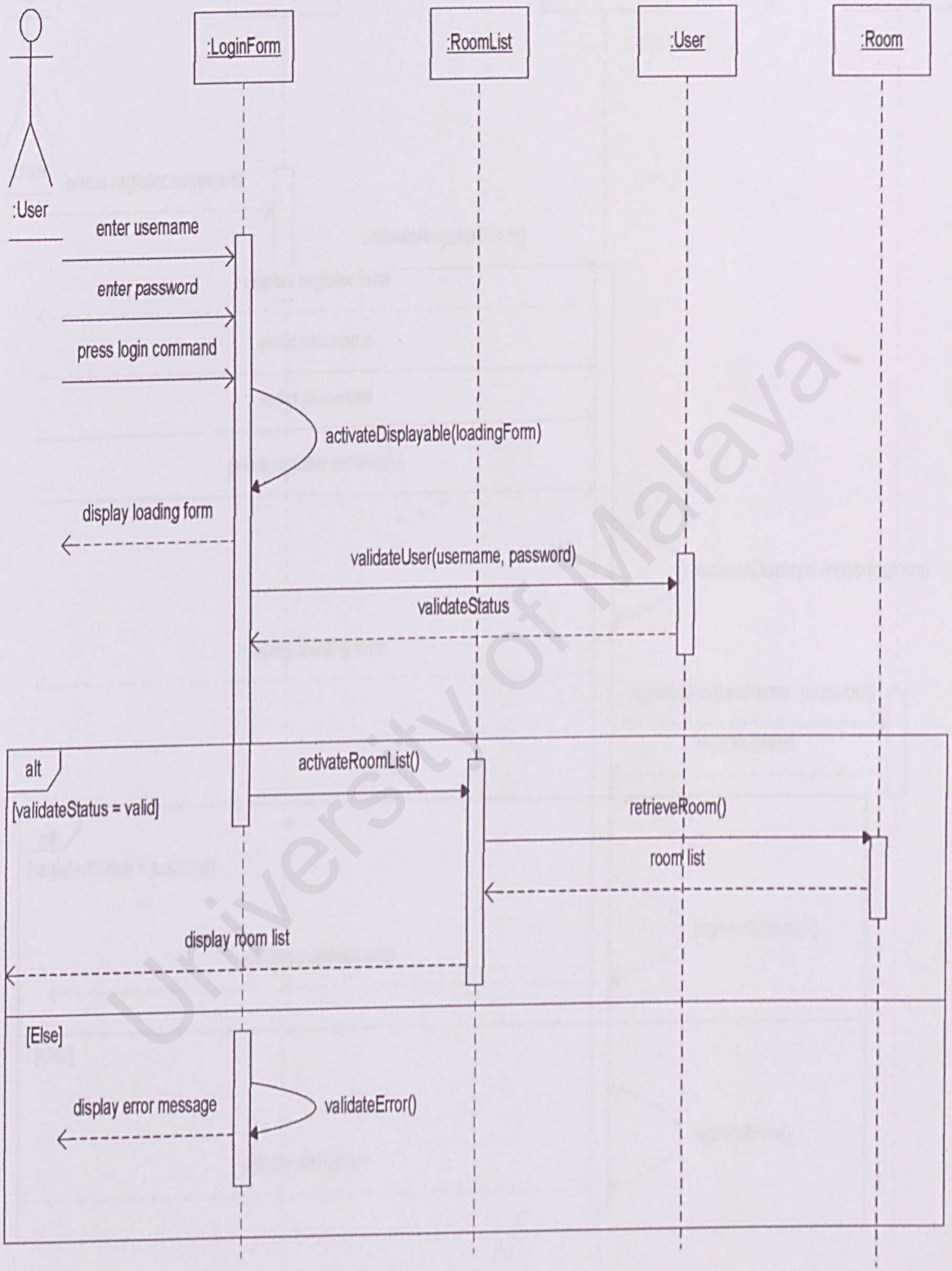


Figure 6.5: Login Sequence Diagram

6.5.2 Register

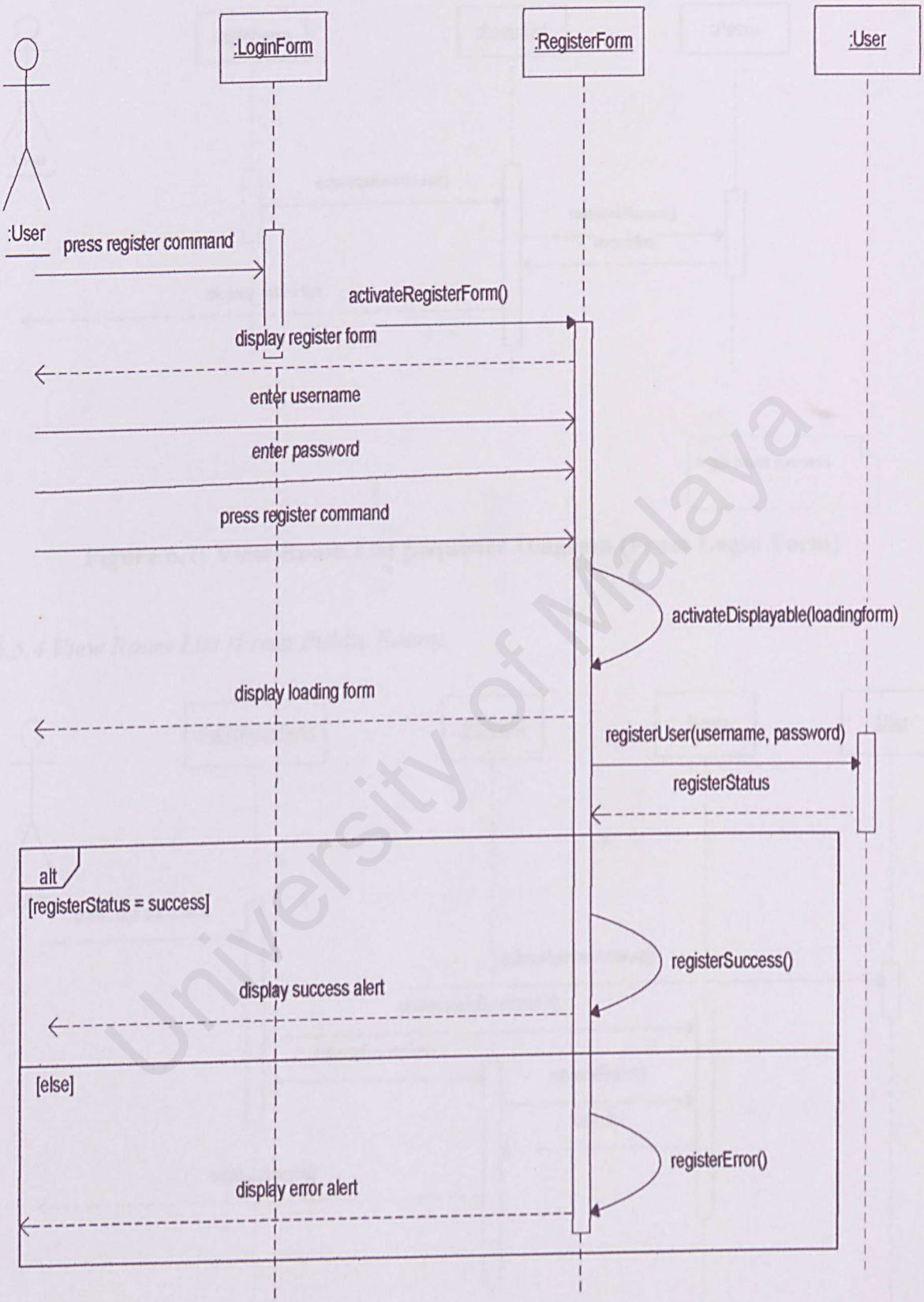


Figure 6.6: Register Sequence Diagram

6.5.3 View Room List (From Login Form)

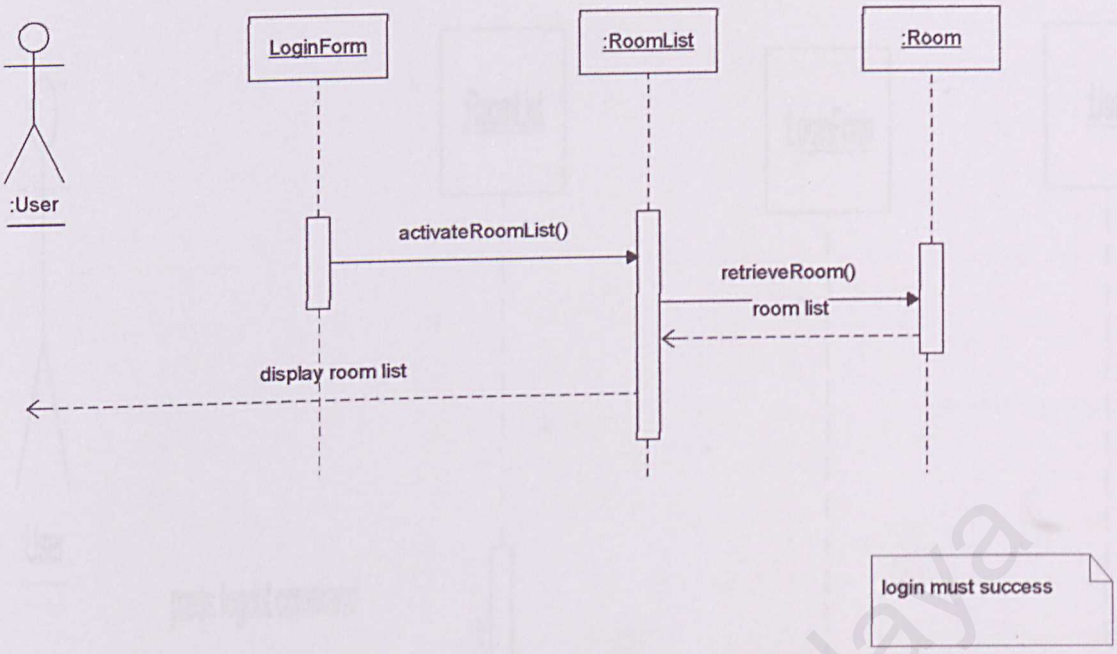


Figure 6.7: View Room List Sequence Diagram (From Login Form)

6.5.4 View Room List (From Public Room)

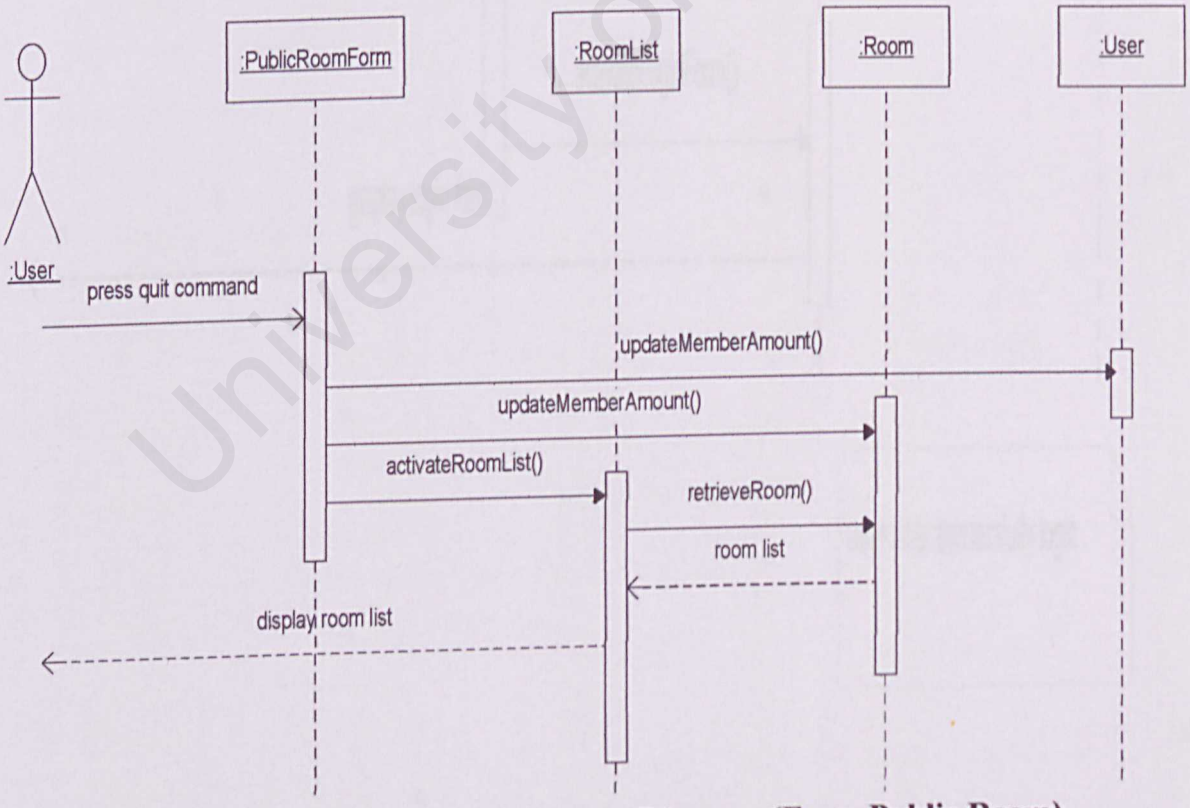


Figure 6.8: View Room List Sequence Diagram (From Public Room)

6.5.5 Logout

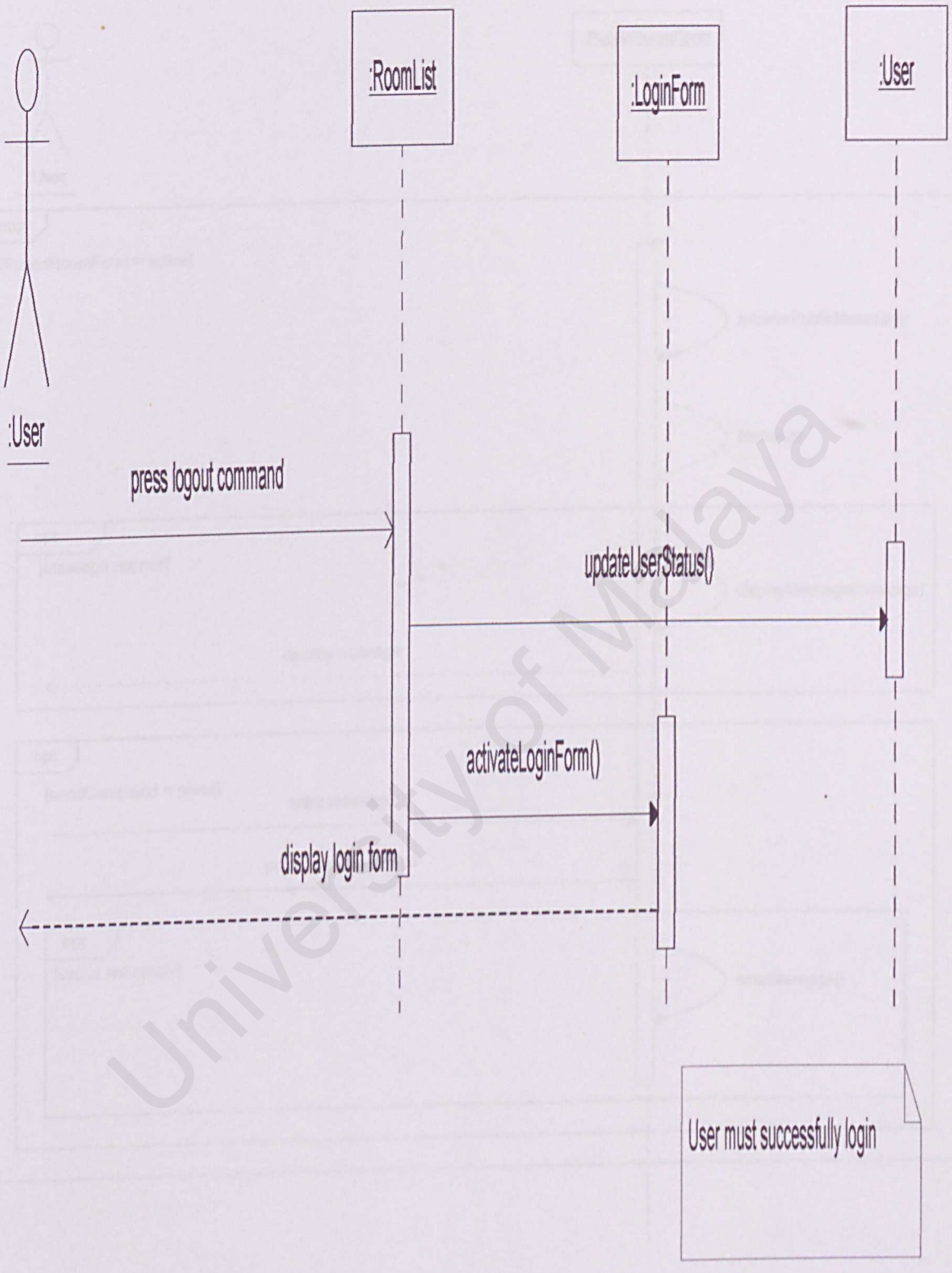


Figure 5.9: Logout Sequence Diagram

6.5.6 Send and Receive Message

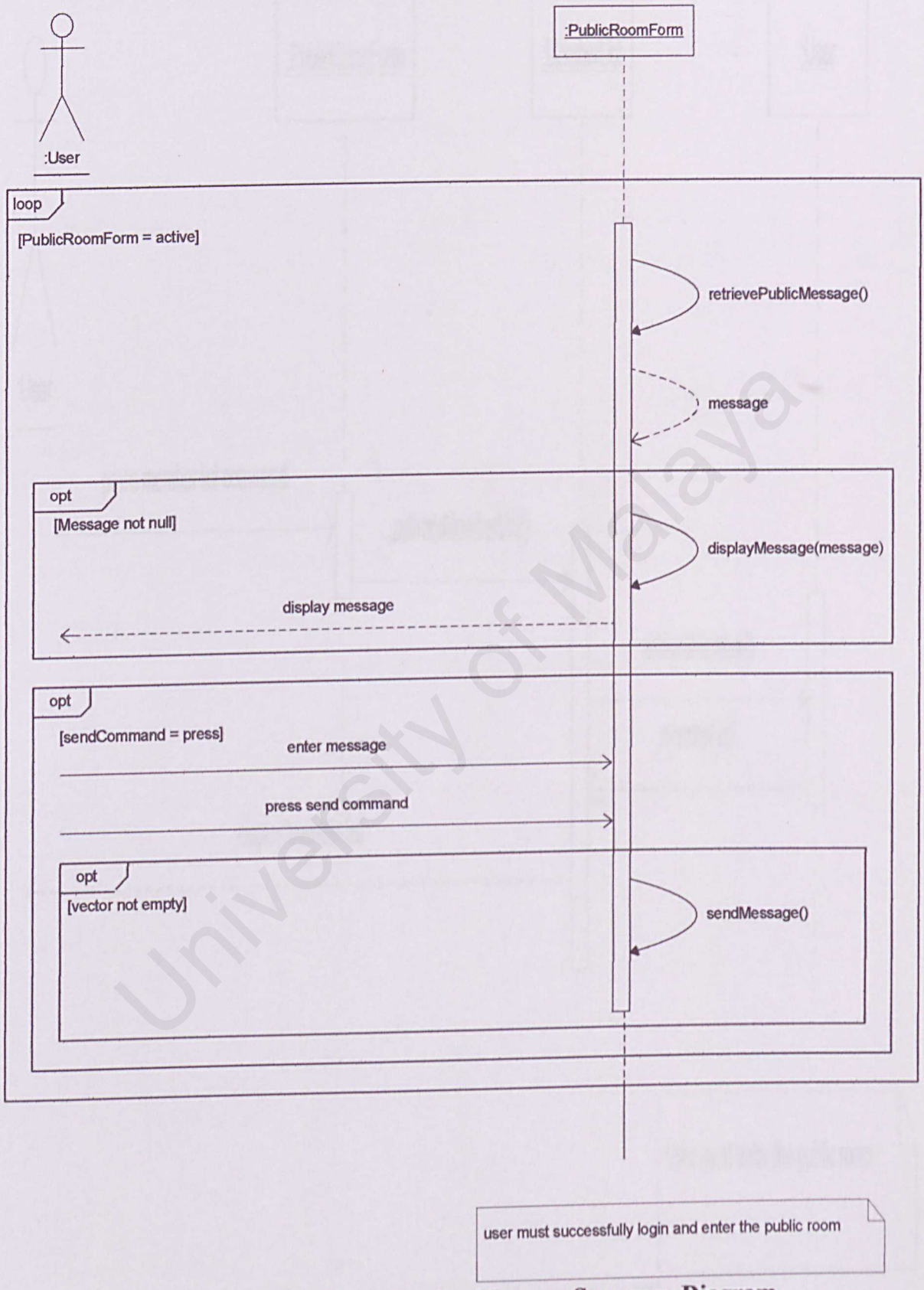


Figure 6.10: Send and Receive Message Sequence Diagram

6.5.7 View Member list

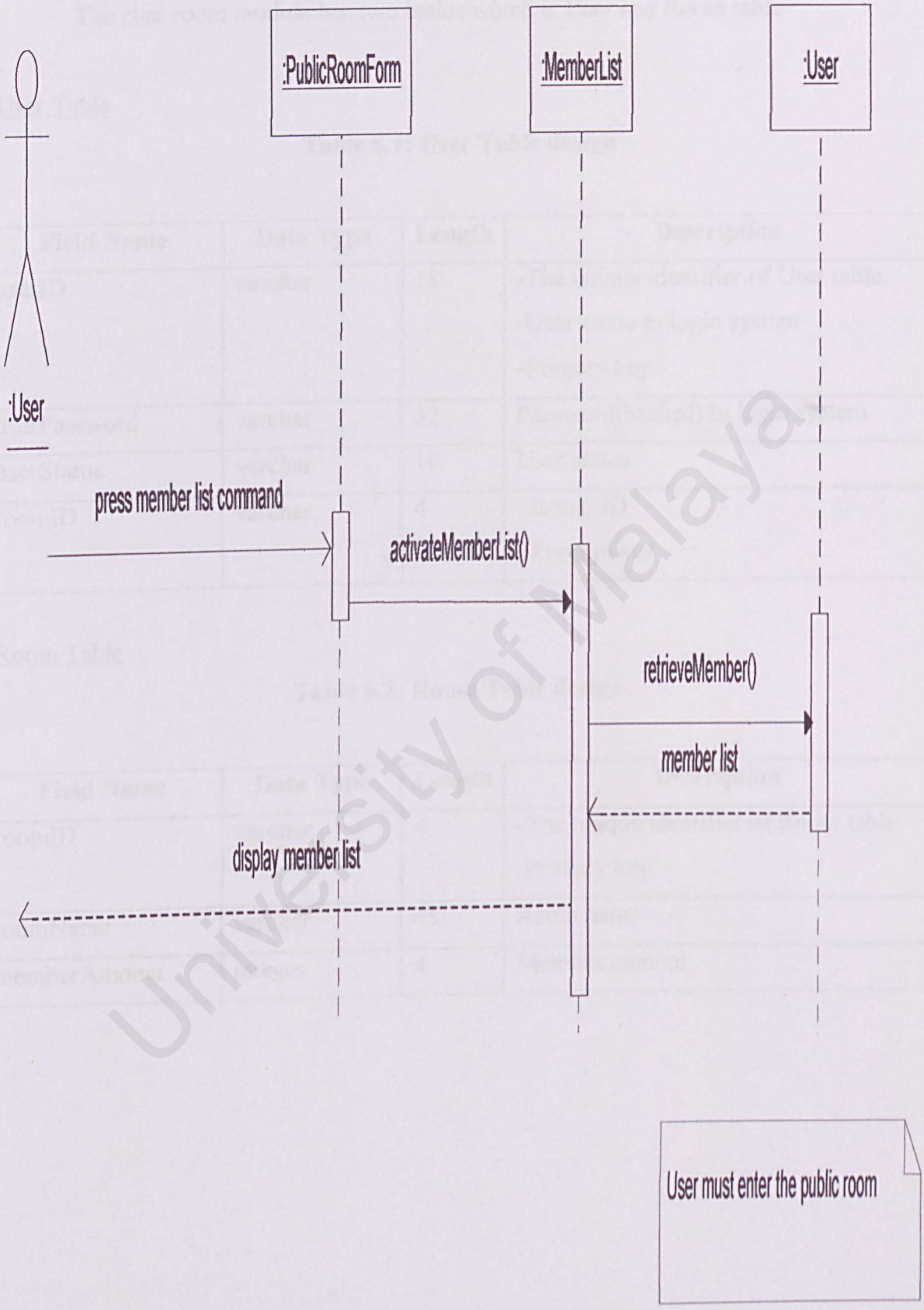


Figure 6.11: View Member List Sequence Diagram

6.6 DATA DESIGN

The chat room module has two tables which is User and Room table.

User Table

Table 6.1: User Table design

Field Name	Data Type	Length	Description
userID	varchar	18	-The unique identifier of User table. -User name to login system -Primary key.
userPassword	varchar	32	Password(hashed) to login system
userStatus	varchar	10	User status
roomID	varchar	4	- Room ID - Foreign key

Room Table

Table 6.2: Room Table design

Field Name	Data Type	Length	Description
roomID	varchar	4	-The unique identifier of Room table. -Primary key.
roomName	varchar	45	Room name
memberAmount	integer	4	Member amount

7.0 SYSTEM DEVELOPMENT

Mobile UM Map is developed using J2ME in developing the mobile client, Servlet in developing the server, and data is stored in the MySQL database.

7.1 DATABASE DEVELOPMENT

For the chat room module in the Mobile UM Map, there have two tables in the MySQL database that stored in the database named mum. Figure 7.1 shows the layout of the User table and Figure 7.2 shows the Room table in the MySQL Administrator.

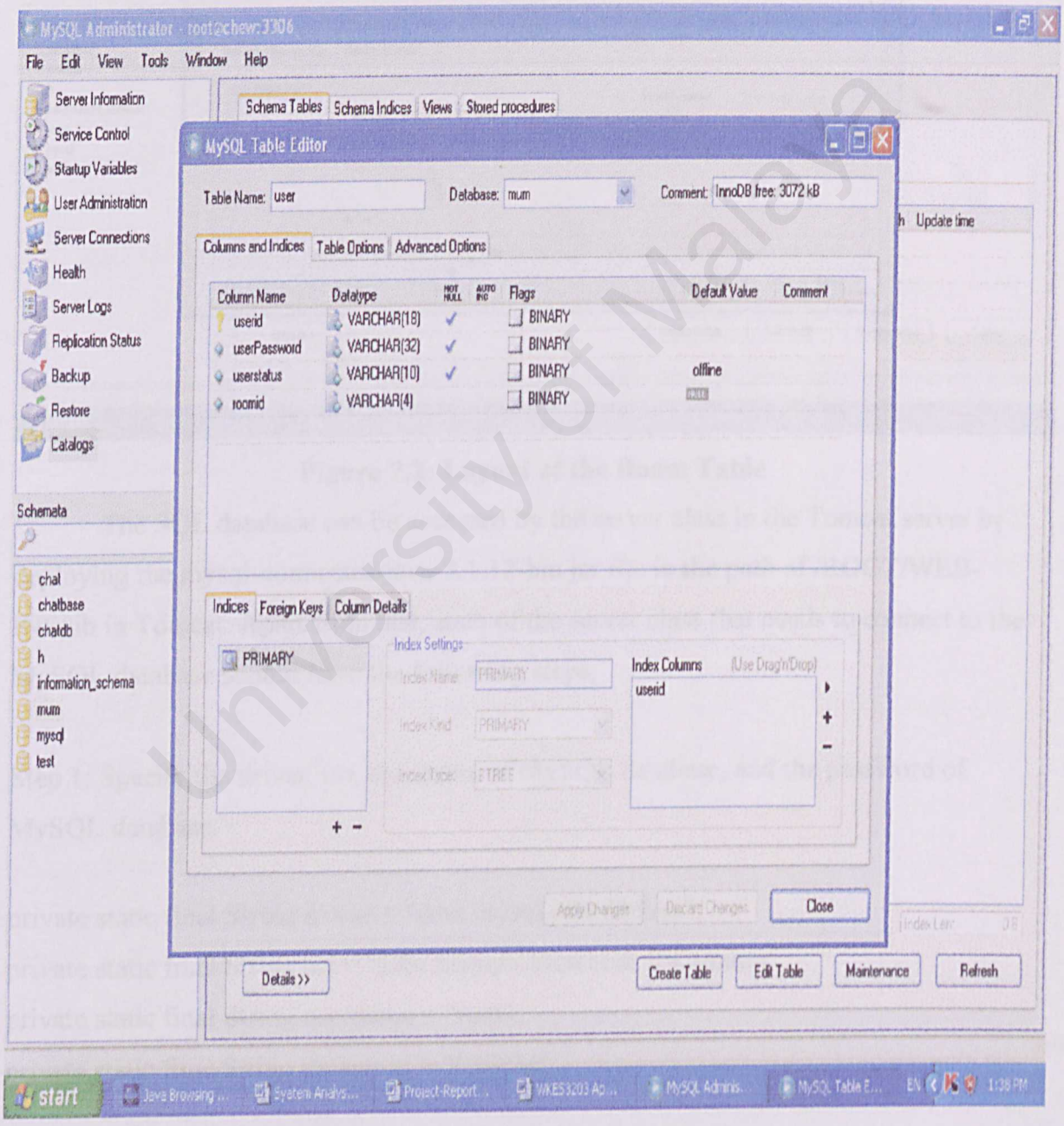


Figure 7.1: Layout of the User Table

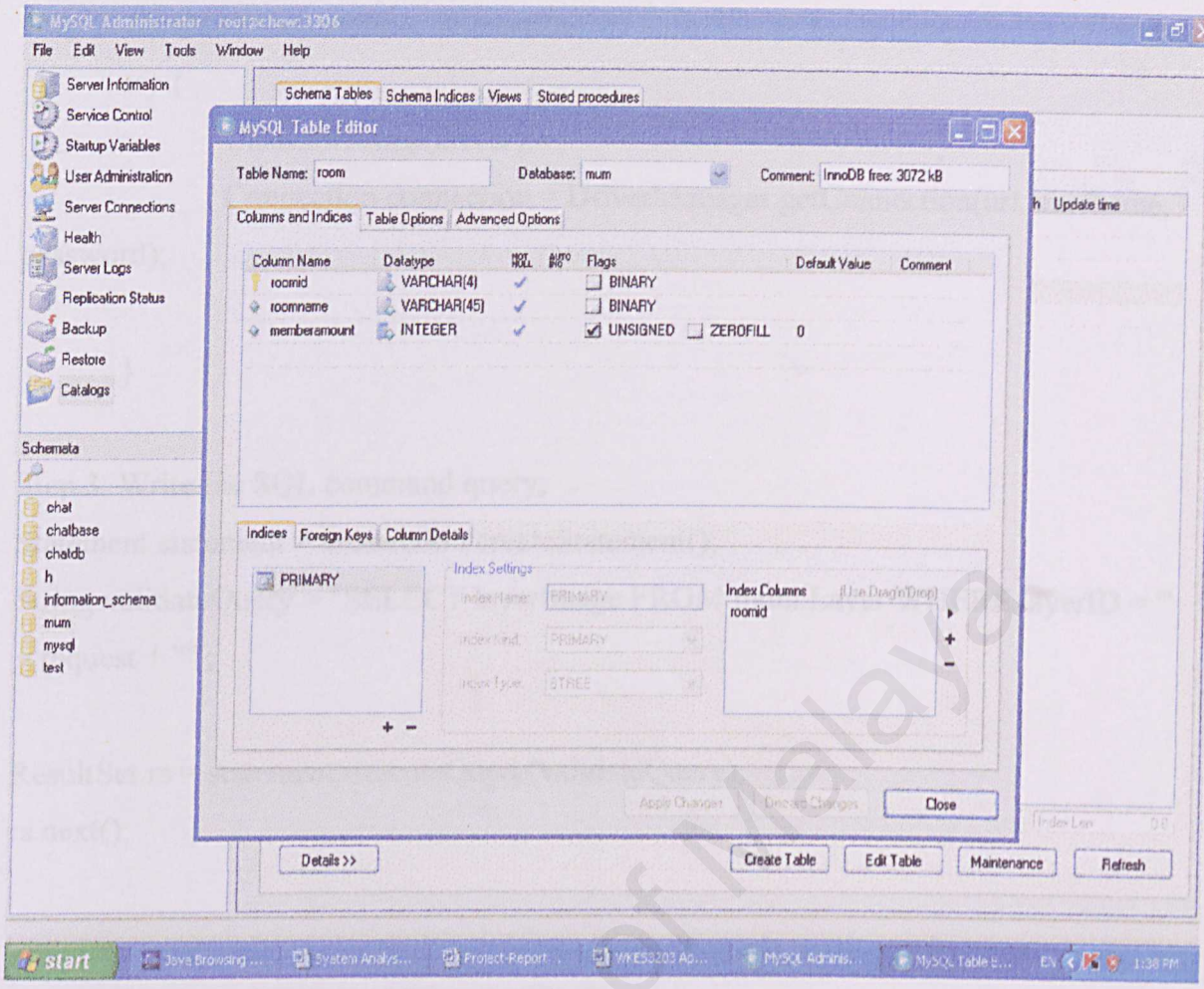


Figure 7.2: Layout of the Room Table

The SQL database can be accessed by the server class in the Tomcat server by deploying the mysql-connector-java-3.1.12-bin jar file in the path of /ROOT/WEB-INF/lib in Tomcat. Apart from that, each of the server class that needs to connect to the MySQL database should have the following steps:

Step 1: Specify the driver, url, username of MySQL database, and the password of MySQL database.

```
private static final String driver = "com.mysql.jdbc.Driver";
private static final String url = "jdbc:mysql://localhost:3306/mum";
private static final String username = "root";
private static final String password = "030034";
```

Step 2: Establish the database connection.

```
try {
        Class.forName(driver);
        Connection connection = DriverManager.getConnection(url, username,
password);
        ..........
}
```

Step 3: Write the SQL command query.

```
Statement statement = connection.createStatement();
String validateQuery = "SELECT layerImage FROM mum.Layer WHERE layerID = '"
+ request + "'";

ResultSet rs = statement.executeQuery(validateQuery);
rs.next();
```

7.2 MOBILE CLIENT DEVELOPMENT

Before need to develop the J2ME application using the Eclipse, there is a lot of setting need to be done. The following is the steps to set the J2ME environments in the Eclipse.

Step1: Install the J2ME Wireless Toolkit into the Eclipse. After that, install the Nokia Series 60 Emulator as well. Figure 6.3 shows the components that already install into the Eclipse at the Platform Components that can be reached by selecting the Preference under the Window menu bar.

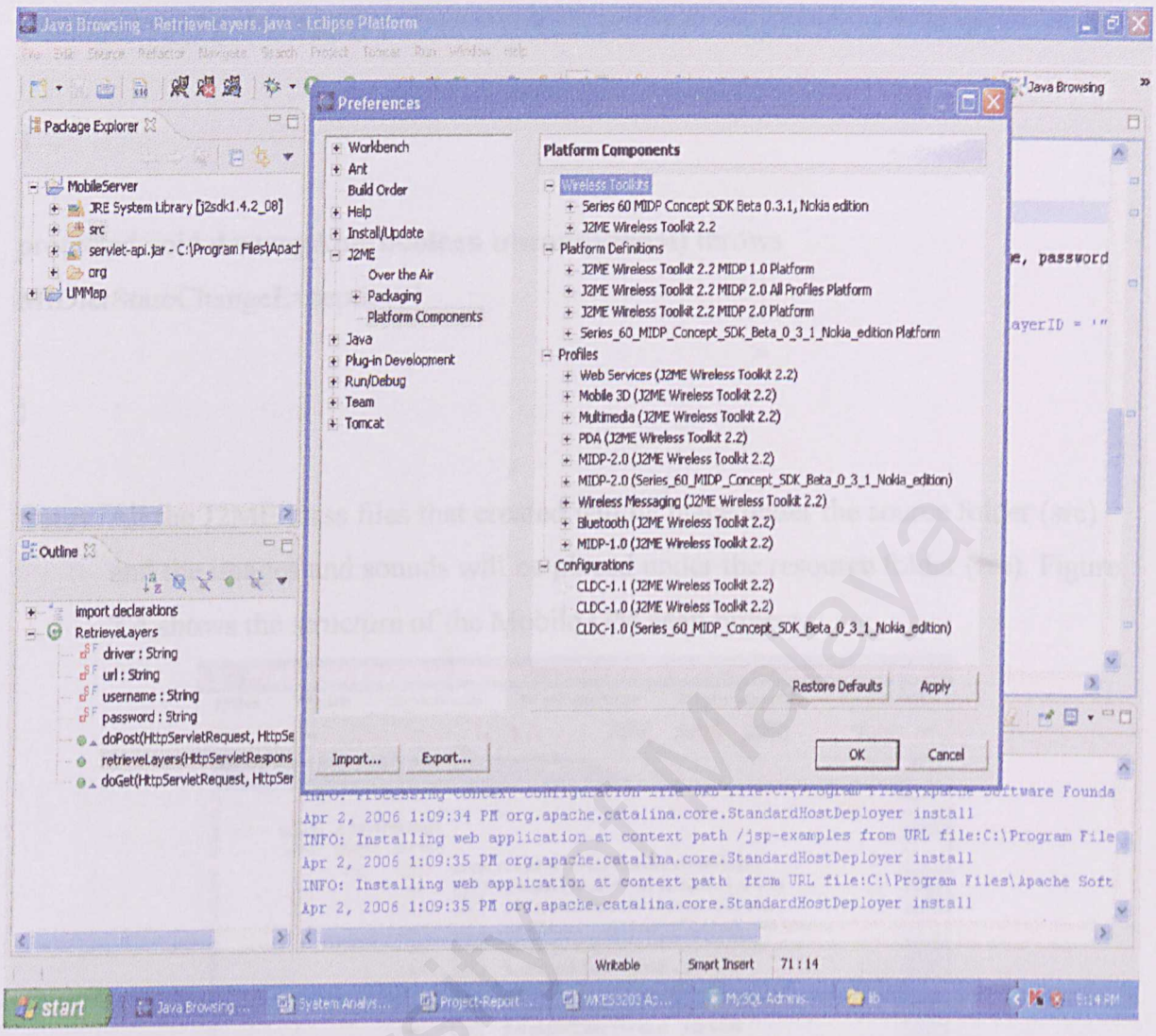


Figure 7.3: List of components of J2ME Wireless Toolkit and Nokia Series 60 Emulator device

Step2: Create the J2ME project. After that, create the Midlet class. The Midlet class will have the components such as startApp(), pauseApp, destroyApp() as shown as the following .

```
protected void startApp() throws MIDletStateChangeException {
        activateDisplayable(currentDisplay);
        mainMenu = new MainMenu(this);
}
```

protected void **pauseApp()** {

}

protected void **destroyApp(boolean unconditional)** throws MIDletStateChangeException {

}

Notes: All the J2ME class files that created will be place under the source folder (src) and the images and sounds will be placed under the resource folder (res). Figure 7.4 shows the structure of the Mobile UM Map project.

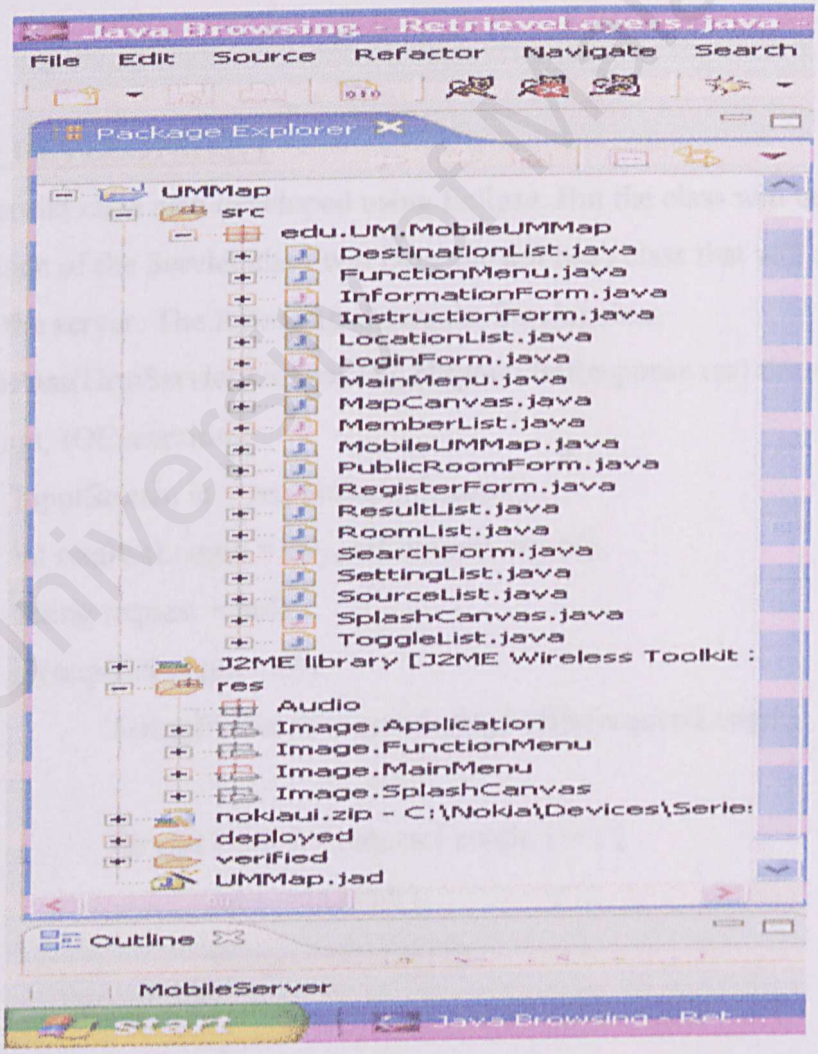


Figure 7.4: Structure of the J2ME mobile client for Mobile UM Map

Step 3: Every J2ME class that need to be connected to the Servlet class in the Tomcat has the following connection string.

```
httpConnection = (HttpConnection)
Connector.open("http://localhost:8080/updateUserStatus");
httpConnection.setRequestProperty("Connection", "close");
httpConnection.setRequestMethod(HttpConnection.POST);

out = httpConnection.openOutputStream();

String request = theMidlet.getChatterName();
for (int i = 0; i < request.length(); i++)
        out.write(request.charAt(i));
```

7.3 SERVER DEVELOPMENT

The Servlet class also developed using Eclipse. But the class will be placed into the Tomcat. Each of the Servlet class will have the doPost() class that will act as the entry point of the server. The following shows the Servlet class.

```
public void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
        InputStream in = req.getInputStream();
        int requestLength = req.getContentLength();
        String request = null;
        if (requestLength > 0){
                StringBuffer sb = new StringBuffer(requestLength);

                for (int i = 0; i < requestLength; i++) {
                        int c = in.read();
                        sb.append((char) c);
                }
```

```
            request = sb.toString();
        }
```

In the Tomcat, there has a web.xml need to be configured in order to specify the connection path that the J2ME client connects to the Tomcat server.

Before the application is running, the Tomcat needs to be started first. Then, the mobile client will be able to send request to the Tomcat server to process the requests and response the appropriate results to the client.

7.4 CODE EXAMPLE

Each of the coding that implemented in the chat room module has the following practices:

1) Every different implementation is in different functions rather than just putting all the implementations into one function. This is for the ease of future maintenance.
2) Put the comment at the certain code line where necessary.
3) Using meaningful class name and variable name.

The following show the way of transformation from the design to the implementation for the register module (client side) in chat room system.

```
package edu.UM.MobileUMMap;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
```

```
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Gauge;
import javax.microedition.lcdui.StringItem;
import javax.microedition.lcdui.TextField;
import javax.microedition.lcdui.Ticker;

public class RegisterForm extends Form implements CommandListener, Runnable {
        private MobileUMMap theMidlet;
        private Ticker ticker;
        private TextField usernameTextField; 1. Meaningful variable name
        private TextField passwordTextField;
        private Command registerCommand;
        private Command cancelCommand;
        private Ticker tickerForm;
        private Form loadingForm;
        private Gauge loadingGauge;
        private StringItem loadingStringItem;
        private Alert nullAlert;
        private String registerString;
        private HttpConnection httpConnection = null;
        private InputStream in = null;
        private OutputStream out = null;
        private Alert registerUserAlert;
        private Alert successAlert;

        2. Meaningful class name
        public RegisterForm(MobileUMMap theMidlet) {
                super("Register");
                this.theMidlet = theMidlet;
```

```
            ticker = new Ticker("Enter your username and password:");
            setTicker(ticker);

            usernameTextField = new TextField("Username:", "", 18,
TextField.ANY);
            passwordTextField = new TextField("Password:", "", 18,
TextField.PASSWORD);
            append(usernameTextField);
            append(passwordTextField);

            registerCommand = new Command("Register", Command.OK, 1);
            cancelCommand = new Command("Cancel", Command.CANCEL, 2);
            addCommand(registerCommand);
            addCommand(cancelCommand);
            setCommandListener(this);

            tickerForm = new Ticker("Registering");

            loadingForm = new Form("Registering");
            loadingGauge = new Gauge("Registering", false, 100, 0);
            loadingStringItem = new StringItem(null, null);

            loadingForm.setTicker(tickerForm);
            loadingForm.append(loadingGauge);
            loadingForm.append(loadingStringItem);
    }

    public void commandAction(Command command, Displayable displayable) {
            if (displayable == this) {
                    if (command == registerCommand) {
                            if (usernameTextField.getString().length() == 0 &&
```

```
passwordTextField.getString().length() == 0) {
                    nullAlert = new Alert("Error !", "Enter your
username, password !", null, AlertType.WARNING);
                    MobileUMMap.activateAlert(nullAlert, this);
                    nullAlert = null;
                }

                else if (usernameTextField.getString().length() == 0) {
                    nullAlert = new Alert("Error !", "Enter your
username !", null, AlertType.WARNING);
                    MobileUMMap.activateAlert(nullAlert, this);
                    nullAlert = null;
                }

                else if (passwordTextField.getString().length() == 0) {
                    nullAlert = new Alert("Error !", "Enter your
password !", null, AlertType.WARNING);
                    MobileUMMap.activateAlert(nullAlert, this);
                    nullAlert = null;
                }

                else {
                    MobileUMMap.activateDisplayable(loadingForm);

                    registerString = usernameTextField.getString() + "
" + passwordTextField.getString();
                    Thread t = new Thread(this);
                    t.start();
                }
            }
```

```
                else if (command == cancelCommand) {
                        usernameTextField.setString("bbbbb");
                        passwordTextField.setString("");
                        theMidlet.activateLoginForm();
                }
        }
}
public void run() {
        String registerResult;
        registerResult = registerUser();
        3. call the function to implement the user registration

        if (registerResult.startsWith("500 REGISTERED")) {
                usernameTextField.setString("");
                passwordTextField.setString("");

                registerSuccess();
        }

        else if (registerResult.startsWith("501 HELD_BY_SOMEONE"))
                registerError("Fail !", "Sorry, the username is held by someone
already, please register with another username.");

        else if (registerResult.startsWith("401 ERROR"))
                registerError("Fail !", "Loading Driver !");

        else if (registerResult.startsWith("402 ERROR"))
                registerError("Fail !", "With Connection !");

        else if (registerResult.startsWith("403 BAD REQUEST"))
                registerError("Fail !", "Bad Request !");
```

```
            else if (registerResult.startsWith("555 MD5_NOT_AVAILABLE"))
                registerError("Fail !", "Algorithm Not Available !");

            else
                registerError("Fail!", registerResult);
        }
//function to implement the user registration
        public String registerUser() {
            try {
                loadingGauge.setValue(10);
                loadingStringItem.setText("    registering: 10%");
        //open the connection    4. Comments
                httpConnection = (HttpConnection)
Connector.open("http://localhost:8080/registerUser");
                httpConnection.setRequestProperty("Connection", "close");
                httpConnection.setRequestMethod(HttpConnection.POST);

                loadingGauge.setValue(20);
                loadingStringItem.setText("    registering: 20%");

                out = httpConnection.openOutputStream();

                loadingGauge.setValue(30);
                loadingStringItem.setText("    registering: 30%");

                for (int i = 0; i < registerString.length(); i++)
                    out.write(registerString.charAt(i));

                loadingGauge.setValue(40);
                loadingStringItem.setText("    registering: 40%");
```

```
                in = httpConnection.openInputStream();

                loadingGauge.setValue(50);
                loadingStringItem.setText("    registering: 50%");

                if (httpConnection.getResponseCode() ==
HttpConnection.HTTP_OK) {
                    int contentLength = (int) httpConnection.getLength();

                    loadingGauge.setValue(60);
                    loadingStringItem.setText("    registering: 60%");

                    if (contentLength == -1)
                        contentLength = 255;

                    loadingGauge.setValue(70);
                    loadingStringItem.setText("    registering: 70%");

                    StringBuffer response = new StringBuffer(contentLength);

                    loadingGauge.setValue(80);
                    loadingStringItem.setText("    registering: 80%");

                    for (int i = 0; i < contentLength; i++)
                        response.append((char) in.read());

                    loadingGauge.setValue(90);
                    loadingStringItem.setText("    registering: 90%");

                    String r = response.toString();
```

```
                    loadingGauge.setValue(100);
                    loadingStringItem.setText("    registering: 100%");

                    return r;
                }

                else
                    return
Integer.toString(httpConnection.getResponseCode());
            }

            catch (Exception e) {
                e.printStackTrace();
                System.out.println("RegisterForm: registerUser");

                return e.toString();
            }

            finally {
                if (httpConnection != null) {
                    try {
                        httpConnection.close();
                        httpConnection = null;
                    }

                    catch (IOException ioe) {
                        ioe.printStackTrace();
                        System.out.println("RegisterForm: registerUser:
httpConnection");
                    }
```

```
                }

                if (out != null) {
                    try {
                        out.close();
                        out = null;
                    }

                    catch (IOException ioe) {
                        ioe.printStackTrace();
                        System.out.println("RegisterForm: registerUser:
out");
                    }
                }

                if (in != null) {
                    try {
                        in.close();
                        in = null;
                    }

                    catch (IOException ioe) {
                        ioe.printStackTrace();
                        System.out.println("RegisterForm: registerUser:
in");
                    }
                }
            }
        }

    public void registerSuccess() {
```

```
            successAlert = new Alert("Success !", "Congratulation, you are
successful registered as a member of chat room in the Mobile UM Map.", null,
AlertType.INFO);
            MobileUMMap.activateAlert(successAlert, this);
            successAlert = null;
        }

        public void registerError(String title, String alertText) {
            registerUserAlert = new Alert(title, alertText, null,
AlertType.WARNING);
            MobileUMMap.activateAlert(registerUserAlert, this);
            registerUserAlert = null;
        }
}
```

The programming technique that used in developing the Mobile UM Map is object-oriented method.

In the J2ME application, the Midlet class will act as the driver to activate the other object class in implementing the particular tasks.

The following code is the Midlet class of the Mobile UM Map that shows the way in activating the other object classes.

```
package edu.UM.MobileUMMap;

import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Image;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

public class MobileUMMap extends MIDlet {     Midlet class
```

```
    private static MobileUMMap theApp;
    private SplashCanvas splashCanvas;
    private MainMenu mainMenu;
    private LoginForm loginForm;
    private RegisterForm registerForm;
    private Displayable currentDisplay;
    private Displayable previousDisplay;
    private RoomList roomList;
    private PublicRoomForm publicRoomForm;
    private MemberList memberList;

    public MobileUMMap() {
            super();
            theApp = this;

            splashCanvas = new SplashCanvas(this);
            currentDisplay = splashCanvas;
    }

    public static MobileUMMap getApp() {
            return theApp;
    }

    protected void startApp() throws MIDletStateChangeException {
            activateDisplayable(currentDisplay);

            mainMenu = new MainMenu(this);
    }
protected void pauseApp() {

}
```

```
    protected void destroyApp(boolean unconditional) throws
MIDletStateChangeException {

    }
    public void close() {
        while (true) {
            try {
                destroyApp(true);
                notifyDestroyed();
                break;
            }

            catch (MIDletStateChangeException e) {
                e.printStackTrace();
                System.out.println("MobileUMMap: close");
            }
        }
    }
    public void setPreviousDisplayable(Displayable previousDisplay) {
        this.previousDisplay = previousDisplay;
    }
    public Displayable getPreviousDisplayable() {
        return previousDisplay;
    }
    public static void activateDisplayable(Displayable displayable) {
        while (true) {
            try {
                Display.getDisplay(getApp()).setCurrent(displayable);
                break;
            }
```

```
                catch (Exception e) {
                    e.printStackTrace();
                    System.out.println("MobileUMMap:
activateDisplayable");
                }
            }
        }
        public static void activateAlert(Alert alert, Displayable fallBackTo) {
            while (true) {
                try {
                    Display.getDisplay(getApp()).setCurrent(alert,
fallBackTo);
                    break;
                }

                catch (Exception e) {
                    e.printStackTrace();
                    System.out.println("MobileUMMap: activateAlert");
                }
            }
        }

        public void setGauge(int currentGauge) {
            splashCanvas.setGauge(currentGauge);
        }


        public void activateMainMenu() {
            if (mainMenu == null)
                mainMenu = new MainMenu(this);
```

```
            currentDisplay = mainMenu;
            activateDisplayable(mainMenu);
    }

    public Image getIconImage(int i) {
            if (mainMenu != null)
                    return mainMenu.getIconImage(i);

            return null;
    }

    public String getChatterName() {
            if (loginForm != null)
                    return loginForm.getChatterName();

            return null;
    }
    Activate login form
    public void activateLoginForm() {
            if (loginForm == null)
                    loginForm = new LoginForm(this);

            loginForm.loadUsername();
            currentDisplay = loginForm;
            activateDisplayable(loginForm);
    }
    Activate register form
    public void activateRegisterForm() {
            if (registerForm == null)
                    registerForm = new RegisterForm(this);
```

```
            currentDisplay = registerForm;
            activateDisplayable(registerForm);
    }
    Activate room list
    public void activateRoomList() {
            if (roomList == null)
                    roomList = new RoomList(this);

            roomList.activateConnection();
            currentDisplay = roomList;
            activateDisplayable(roomList);
    }

    public String getRoomString() {
            if (roomList != null)
                    return roomList.getRoomString();

            return null;
    }
    Activate public room form
    public void activatePublicRoomForm() {
            if (publicRoomForm == null)
                    publicRoomForm = new PublicRoomForm(this);

            publicRoomForm.activateConnection();
            currentDisplay = publicRoomForm;
            activateDisplayable(publicRoomForm);
    }
    Activate member list
    public void activateMemberList() {
            if (memberList == null)
```

```
                memberList = new MemberList(this);

                memberList.activateConnection();
                currentDisplay = memberList;
                activateDisplayable(memberList);
        }
}
```

8.0 TESTING TECHNIQUE

8.1 COMPONENT TESTING

When developing the chat room module, the components testing will be carried out to ensure that each of the components are in expected results when the application is given appropriate inputs and there will have the alert or exception catching whenever the inappropriate input is enter into the application.

The following shows the test data of:

1) Login module

Table 8.1: Login module test data

Test Case No.	Username field	Password field	Result
1	Valid username	Valid password	Login success
2	Valid username	Invalid password	Unauthorized
3	Invalid username	Valid password	Unauthorized
4	Invalid username	Invalid password	Unauthorized
5	-	Valid password	Display error message
6	Valid username	-	Display error message
7	-	-	Display error message

2) Registration module

Table 8.2: Test data for Registration module

Test Case No.	Username field	Password field	Result
1	Not exist	Any characters	Register success
2	Exist	Any characters	Register unsuccessful
3	-	-	Display error message

4	-	Any characters	Display error message
5	Any characters	-	Display error message

3) Public chat room module

Table 8.3: Test data for chat room module

Test Case No.	Text Field	Result
1	Any characters	Message received by all members
2	Null	No action

8.2 SYSTEM TESTING

After all the components had been integrated, the application was tested using emulator. The application also has been tested using the real phone. As the map consumes a lot of memory, the performance using the real phone was not satisfying.

9.0 DISCUSSION

9.1 PROBLEMS AND SOLUTIONS

The difficulties in developing the chat room module is ensuring that the mobile client will keep listening to the message that sent by the senders in the same room. The solution to this problem is creating the Thread and keep the thread running every half second while the application still alive.

9.1 USER EVALUATION

During the Malaysia Technology Expo (MTE), the visitors had been allowed to test the system using the emulator. According to the visitors' comments, the system is easy to use.

9.3 POSITIVE AND NEGATIVE FEATURES

Every system developed has its own positive and negative features. The chat room module also has its own pros and cons.

9.3.1 Positive features

The chat room module in the Mobile UM Map is easy to use. It prepares a set of rooms that allow the users to choose. In the future maintenance, the room can be added by updating the Room table in the MySQL database.

The chat room is the public room which will allow multi-users to enter and having discussion among the UM students regarding the current issues or activities that occurs within the campus.

9.3.2 Negative features

The chat room module is not allowing the users to create the room they want. They only can choose from the system that already being set. It also not allows the users to have the private messaging with their own friends.

9.4 FUTURE ENGHANCEMENTS

In the future enhancements, the chat room can be added extra functions which is video conference. The images will be catch up and store into the array in byte. After that, the images will be retrieve and display to the screen accordingly.

The system also can be extending to have the private room for each of the members. The member in a particular room can be selected by the user and then can send the message to that particular selected member without view by other members.

10.0 CONCLUSION

The Mobile UM Map is a useful application to let the UM students and staffs to familiarize the UM environments. Not only they can search the path from one location to another location, they also can carry out the discussion among the UM students regarding any issues occurring within campus. This will keep the students to be alert on the current issue and know what the latest activities that held within the campus are.

The chat room module has fulfill some of the objectives which is encourage knowledge sharing, act as mobile counseling, and allow the UM students and staffs to feel that they are part of the member in the UM as they can contribute their opinions in regarding a particular topic or activity that organized by certain clubs or students society by using the chat room application in the Mobile UM Map.

In the Mobile UM Map, there involves three part of the subsystem, which is mobile client, server and database. In order to use the Mobile UM Map application, the user must have a mobile phone which fulfills the minimum requirements which is Nokia Series 60. After the user has successfully deployed the application into the mobile phone, then the user can use the application. The mobile client will connect to the Tomcat server using http connection in order to retrieve the appropriate response from the server. The server also connects to the MySQL database in order to retrieve some of the data to send to the mobile client. As long as the Tomcat server which located at a computer is started, the Mobile UM Map application can be used at anytime and anywhere. The server also required the frequent maintenance to ensure its continuity of services.

In developing the chat room module, I had improved my programming skills in J2ME, Servlet, and MySQL. I also had followed the good programming practices, such as using the comments and use of meaningful variable declaration. I also follow the appropriate methodology which is the spiral model. Each of the function that developed will be evaluated its risk in terms of the possibilities that the function will be successfully developed.

In a nutshell, throughout the project I had gained the valuable experience and knowledge in developing the J2ME application.

11.0 REFERENCES

1) Wassim Itani, &Ayman Kayssi. (January 2004). J2ME application-layer end-to-end security for m-commerce. Journal of Network and Computer Applications, 27, 13-32. Obtained July 26, 2005, from ScienceDirect database.

2) Anup K. Ghosh, & Tara M. Swaminatha. (February 2001). Software Security And Privacy Risks In Mobile E-commerce. Journal of Communications of the ACM, 44, 51-57. Obtained August 24, 2005, from ACM database.

3) Marc Smith, JJ Cadiz, & Byron Burkhalter. (2000, December). Conversation Trees and Threaded Chat. Whitepaper presented in Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, United States.

4) Mobile Lifestyle. (2005). Obtained August 24, 2005, from http://www.viztel.com/products_buddytalk.htm

5) Mobile Chat. (2004). Obtained August 24, 2005, from http://myxmobile.com/index.php?mwop=Mobile%20Chat&myProjectSess=2e7590843a8f3418a27f99f87dd0f734

6) ddMobile. (2003). Obtained August 23, 2005, from http://www.ddmobile.com

7) Mocha.(n.d). Obtained August 18, 2005, from http://www.zgroup-mobile.com/published_games/application/mocha/mocha.html

8) Mobile Chat and Instant Messaging service. (July 2003). Obtained September 11, 2005, from http://www.mtc-vodafone.com/m-chat/SMS.Help.pdf.

9) Alistair Cockburn, & Laurie Williams. (2000). The Costs and Benefits of Pair Programming. Taken September 1, 2005, from http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF.

10) Barry Boehm, Wilfred J.Hansen. (January 25, 2001). Uderstanding the Spiral Model as a tool for Evolutionary Acquisition. Taken September 9, 2005, from http://sunset.usc.edu/publications/TECHRPTS/2001/usccse2001-501/usccse2001-501.pdf.

11) Barry W. Boehm. (May 1988). A Spiral Model of Software Development and Enhancement. Taken September 9, 2005, from http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral%20Model%20Boehm.pdf